



TAMPEREEN TEKNILLINEN YLIOPISTO  
TAMPERE UNIVERSITY OF TECHNOLOGY

KARTHHIC RAMACHANDRAN PONNALAGI  
A REDUNDANT ZIGBEE COMMUNICATION MECHANISM FOR  
AVOIDING COLLISION IN A MONORAIL SYSTEM

Master of Science thesis

Examiners:  
Jose L. Martinez Lastra  
Examiner and topic approved by the  
Faculty Council of the Faculty of  
Engineering Sciences on March 2017

## ABSTRACT

### **KARTHHIC RAMACHANDRAN PONNALAGI: A REDUNDANT ZIGBEE COMMUNICATION MECHANISM FOR AVOIDING COLLISION IN A MONORAIL SYSTEM**

Tampere University of technology

Master of Science Thesis, 56 pages, 3 Appendix pages

August 2018

Master's Degree Programme in Automation Engineering

Major: Factory Automation and Industrial Informatics

Examiners: Prof. Jose L. Martinez Lastra,

Keywords: Electric monorail system, Zigbee, Wireless communication, Automatic control, Python, Programmable logic controller, Internet of Things.

The safety of machines plays an important role in an automation system. The accident of transportation system leads to delay or complete blockage of production which leads to loss. In time-sensitive production, it could lead to a defective product. To avoid such accidents collision avoidance mechanisms were implemented. The collision avoidance mechanisms are made of using sensors and by sharing status between themselves.

The monorail system is an overhead transportation system which consists of a rail in which few monorail carriers run. The rail is a closed loop and the carriers run in one direction in the rail. The carriers are controlled by the floor programmable logic controller (PLC) which orchestrates the movement of the carriers through the primary Wi-Fi communication between them. The main task was to implement a collision avoidance mechanism which should act when the Wi-Fi communication between the carrier and the floor PLC fails. This collision avoidance mechanism prototype is done using internet of things (IoT) devices.

The IoT is a fast-growing technology with a purpose of connecting any physical devices into a network. In that network the data from the physical device can be processed or shared with similar devices.

In this thesis the background information of the monorail transfer system is studied, and the major components used are explained. The different methods of wireless communications were checked, and a suitable wireless communication is chosen for the collision avoidance system. Using an IoT device the PLC of the monorail carrier is connected to IoT network which is used to communicate between other carriers. By communicating using the secondary IoT network, the carriers share their position information which results in collision avoidance. In this case not only avoiding collisions but also the carrier is moved to the maintenance position by the IoT device when it blocks the path of the neighboring carrier.

A prototype was built and it was tested on the test platform, and the necessary changes were made to make it fully functional. A simulator is created to show the working logic after the test platform became unavailable. The IoT collision avoidance mechanism can also be used for controlling the carriers when the floor PLC is down or as a test platform to trigger the functions of the carriers. With some additional PLC programs for secondary control this mechanism could also be used as a secondary control to the floor PLC for the carriers.

## PREFACE

எப்பொருள் யார்யார்வாய்க் கேட்பினும் அப்பொருள்  
மெய்ப்பொருள் காண்ப தறிவு

To discern the truth in everything, by whomsoever spoken, is wisdom.

- Thiruvalluvar

My sincere gratitude to my thesis supervisor Prof. Jose L. Martinez Lastra and Cimcorp project manager, Mr. Pasi Kankaanpää for believing in me and giving me this opportunity to do my thesis. I would like to thank Dr., Andrei Lobov, Dr., Jani Jokinen, and Mr. Jani Tuomola for guiding me and helping me to complete the thesis. I would also like to thank Cimcorp Engineers Mr. Matti Pentti, Mr. Paavo Ranta, Mr. Janne Raatikainen for helping me in building and understanding the technical aspects of my thesis, and Mr. Wael Mohammed for reviewing and helping me writing this thesis report.

My heartfelt thanks to all my friends, and my family, for motivating me.

Tampere 08.08.2018

Karthhic Ramachandran Ponnalagi

## CONTENTS

1.	INTRODUCTION .....	1
1.1	PROBLEM STATEMENT .....	2
1.2	RESEARCH QUESTIONS .....	2
1.3	SCOPE .....	3
1.4	LIMITATIONS AND CHALLENGES .....	3
2.	BACKGROUND .....	4
2.1	MATERIAL HANDLING SYSTEM .....	4
2.2	PROGRAMMABLE LOGICAL CONTROLLERS .....	5
2.3	SAFETY SYSTEMS .....	7
2.4	IoT DEVICES .....	10
2.5	WIRELESS COMMUNICATION IN AUTOMATION .....	12
2.5.1	DIFFERENT TYPES OF WIRELESS NETWORKS .....	13
2.5.2	INTRODUCTION TO ZIGBEE .....	14
2.6	SIMILAR WORKS .....	16
3.	METHODOLOGY .....	20
3.1	ACTUAL SYSTEM .....	20
3.2	SELECTION OF CONTROLLER BOARD .....	25
3.2.1	ARDUINO AND SMT32 .....	26
3.2.2	ASUS TINKER BOARD .....	26
3.2.3	BEAGLE BONE .....	26
3.2.4	RASPBERRY PI .....	26
3.3	SELECTION OF WIRELESS DEVICE .....	27
3.4	CHOICE OF PROGRAMMING LANGUAGE .....	27
3.5	REASON FOR USING SIMULATOR .....	28
4.	SYSTEM DESIGN AND IMPLEMENTATION .....	29
4.1	IMPLEMENTATION .....	29
4.2	HARDWARE .....	31
4.2.1	ZIGBEE .....	31
4.2.2	RASPBERRY PI .....	31
4.2.3	PLC .....	32
4.3	SOFTWARE .....	32
4.3.1	THE PROGRAM .....	32
4.3.2	CPPPO LIBRARY FOR PYTHON .....	39
4.3.3	XCUD CONFIGURATION PLATFORM FOR ZIGBEE .....	39
4.3.4	OTHER IoT DEVELOPMENT APPLICATIONS .....	39
4.4	COMMUNICATION .....	41
4.4.1	SETTING UP A ZIGBEE NETWORK .....	41
4.4.2	CONNECTION BETWEEN ZIGBEE AND RASPBERRY PI .....	42
4.4.3	CONNECTION BETWEEN RASPBERRY PI AND PLC .....	44
4.4.4	CONNECTION BETWEEN ZIGBEE AND PLC .....	44
4.5	SIMULATOR .....	44

5.	RESULTS AND DISCUSSION .....	46
5.1	TESTING .....	46
5.2	DISCUSSION .....	49
6.	CONCLUSION AND FUTURE WORK .....	52
6.1	CONCLUSION .....	52
6.2	FUTURE WORK .....	53
7.	REFERENCES.....	54

APPENDIX A: DIN MOUNT FOR THE RASPBERRY PI AND ZIGBEE

APPENDIX B: TAG READ TIME CALCULATION PROGRAM

## LIST OF FIGURES

<i>Figure 1 a) Monorail conveyor in Ford Motor Company and b) typical Conveyor unit [19][20].....</i>	<i>4</i>
<i>Figure 2 Relay logic racks for controlling [13] .....</i>	<i>6</i>
<i>Figure 3 Sample Ladder Program [14] .....</i>	<i>6</i>
<i>Figure 4 Safety priority hierarchy .....</i>	<i>7</i>
<i>Figure 5 Working of a PLS proximity sensor [15] .....</i>	<i>8</i>
<i>Figure 6 Different zones of a PLS proximity sensor [15] .....</i>	<i>9</i>
<i>Figure 7 Laser Distance measurement sensor.....</i>	<i>9</i>
<i>Figure 8 Number of connected devices with forecast [21] .....</i>	<i>11</i>
<i>Figure 9 Revolutions of Industrial development [5].....</i>	<i>11</i>
<i>Figure 10 Architecture of Zigbee.....</i>	<i>15</i>
<i>Figure 11 Configuration and connections between Zigbee devices. [22] .....</i>	<i>15</i>
<i>Figure 12 Digi Mesh Network [22] .....</i>	<i>16</i>
<i>Figure 13 System frame [24] .....</i>	<i>17</i>
<i>Figure 14 Proposed design block diagram.[6] .....</i>	<i>18</i>
<i>Figure 15 System structure diagram [10].....</i>	<i>18</i>
<i>Figure 16 Instruction flow from higher level to lower level .....</i>	<i>21</i>
<i>Figure 17 System components [29] .....</i>	<i>22</i>
<i>Figure 18 90-degree curve of turning of the monorail carrier.....</i>	<i>23</i>
<i>Figure 19 Position sensor with Position Strip [27] .....</i>	<i>24</i>
<i>Figure 20 Monorail Transfer, Typical systems [29].....</i>	<i>24</i>
<i>Figure 21 Curve where laser-based measurement is not helpful .....</i>	<i>25</i>
<i>Figure 22 The prototype.....</i>	<i>27</i>
<i>Figure 23 Proposed implementation.....</i>	<i>29</i>
<i>Figure 24 Stages of implementation .....</i>	<i>30</i>
<i>Figure 25 Zigbee Series 2 Module. [26] .....</i>	<i>31</i>
<i>Figure 26 Raspberry pi [25] .....</i>	<i>32</i>
<i>Figure 27 Zigbee coverage and communication envelop .....</i>	<i>33</i>
<i>Figure 28 Flowchart of the program .....</i>	<i>34</i>
<i>Figure 29 Function to calculate the distance between careers. ....</i>	<i>35</i>
<i>Figure 30 Calculating the distance between carriers at the end of the loop.....</i>	<i>35</i>
<i>Figure 31 UML diagram of the communication .....</i>	<i>37</i>
<i>Figure 32 Functions which communicates with the PLC .....</i>	<i>38</i>
<i>Figure 33 XCTU configuration toolbox window .....</i>	<i>39</i>
<i>Figure 34 Node-Red User Interface showing slide control and output .....</i>	<i>40</i>
<i>Figure 35 Cross connection of serial pins between RPI and xbee .....</i>	<i>42</i>
<i>Figure 36 Connected Xbee and Raspberry pi.....</i>	<i>43</i>
<i>Figure 37 The Raspberry pi Zigbee setup.....</i>	<i>43</i>
<i>Figure 38 Position logic in the simulator .....</i>	<i>44</i>
<i>Figure 39 Simulator Window. ....</i>	<i>45</i>
<i>Figure 40 Test Setup a) Line of sight b) non line of sight.....</i>	<i>46</i>

<i>Figure 41 Throughput test 1-meter test. ....</i>	<i>47</i>
<i>Figure 42 Throughput test 6-meter test ....</i>	<i>47</i>
<i>Figure 43 Range Test 6-meter with obstruction in LOS.....</i>	<i>48</i>
<i>Figure 44 Time taken for the Python program to read the position. ....</i>	<i>49</i>

## LIST OF SYMBOLS AND ABBREVIATIONS

6LowPAN	IPv6 over Low-Power Wireless Personal Area Networks
AMHS	Automated Material Handling System
API	Application Program Interface
ARM	Advanced RISC Machine
AT	Application Transparent
CFT	Continuous Flow Transport
CPPPO	Communication Protocol Python Parser and Originator
DIY	Do it yourself
EMS	Electrified Monorail System
GHz	Giga Hertz
GPIO	General Purpose Inputs and Outputs
GTAI	Germany Trade and Invest
HART	Highway Addressable remote Transducer
HDMI	High-Definition Multimedia Interface
HMI	Human Machine Interface
HTML	Hyper Text Markup Language
IoT	Internet of Things
IR	Infrared
LAN	Local Area Network
LOS	Line of Sight
MAC	Media Access Control
MAN	Metropolitan area Network
MHz	Megahertz
OMV	Overhead Monorail Vehicles
PAN	Personal Area Network
PLC	Programmable Logic Controller



PLS	Proximity Laser Scanner
RPI	Raspberry Pi
SBC	Single Board Computer
SD	Secure Digital
SOC	System On Chip
TTL	Transistor-Transistor Logic
UDP	User Datagram Protocol
UI	User Interface
UML	Unified Modelling Language
USB	Universal Serial Bus
WAN	Wide Area Network
WIFI	Wireless Fidelity
WPAN	Wireless Personal Area Network

# 1. INTRODUCTION

In the modern world of mass production, the compelling need for flexible, expandable, reprogrammable, fast means of transportation. The traditional conveyor belts are suitable for systems, which do not require high-speed transportation due to its speed constraints. It is also space consuming in the production floor and makes it expensive in a mass production factory by consuming lot of space and requiring series of multiple conveyor units to cover the distance. In this case a monorail system which is an overhead transportation system. It has multiple individual carriers in its rail for transporting products to different places in the facility. It replaces the conveyors for high-speed transportation systems. The modern monorail systems are intelligent and fast enough to work as a group and complete complex tasks precisely and share the workload evenly between each other.

The intelligence of any system mainly depends upon how it manages to accomplish its task during any unexpected situations. Moreover, as the system gets more intelligent, it requires more data from its environment through sensors and communicating to other systems in the network. In general, an automation facility has a single server, or several distributed servers controlling the machineries in the production floor. These servers does all the computations and interactions between machines by orchestrating the process in an orderly flow. It is possible to use IoT devices along with the servers. IoT's are network-based devices that communicates and control other compatible devices and machines. They have input and output ports commonly known as IO port and communication ports like serial, Ethernet, Bluetooth and wireless. Similar like the servers, IoT devices can communicate with the PLC using different methods like communicating through network. They also operate by connecting their Inputs and outputs or IOs directly with the IOs of the controller or PLC. The development of the IoT in the recent days paved the way for the creation of the microcomputers with size as small as of a credit card and lesser known as SOC (system on chip). In addition, not only processing the data obtained from the machines it is possible to control them from the IoT devices. IoT devices are not just limited to SOC's but also includes Micro Controllers, Communication modules, and sensor modules. The modules can be of different nature like sensors, wireless modules, converters, signal processing modules, even additional IO ports. Modularity in the design of the devices makes it possible to add a new module or replace an old one with ease.

The communication between the IoT devices are mostly done wirelessly for convenience. There are many wireless protocols available for communication between IoT devices. But the selection of the wireless solution depends mainly upon the range of coverage of the device, environment affecting the propagation of the signal and security of the device. Three main networking methods are used they are:

- Wireless personal area network (PAN)
- Local area network (LAN)
- Wide area network (WAN)

The PAN is a short-range network, LAN is a medium range network and WAN is a long-range network respectively.

This thesis is a development project on an overhead monorail system for an automation specialization company Cimcorp OY. Cimcorp is providing automated material handling solutions for numerous fields like Tire manufacturing, Dairy, postal, Retail etc., The thesis concentrates on the tire manufacturing system's green tire transportation system. An overhead monorail carrier transporting the green tires between different parts of the facility. The monorail carrier will transports the green tire from the tire building machines to the buffer and from the buffer to the curing presses. These monorail carrier systems are high-speed transportation units. They operate in tire manufacturing area, there will be lot of dusts and other kind of physical disturbance like steam, vapors, atomized oil particles. Also the ambient temperature will be high.

## **1.1 PROBLEM STATEMENT**

In a systematically orchestrated automation system with multiple mobile units running in a common rail, disruption due to blockage by one of its own devices leading to production loss. In monorail carrier system with multiple carriers in the overhead rail, if any one of the carriers stops due to any reason, then the next carrier will also halt leading to the stoppage of the process flow. The only way to solve this issue is to manually move the carrier to the maintenance position to clear the traffic for other carriers. Presuming that the stoppage is caused by communication issues, then the carrier cannot be accessed by the HMI to move to the maintenance position, but by using an IR remote and jog it to the maintenance position. Sometimes the carrier might lose its communication with the floor PLC, where the Wi-Fi connectivity gets weak. And in the corners, the current laser sensor cannot sense the neighbouring carrier because the laser sensor can only in a straight line and cannot sense in the corners.

## **1.2 RESEARCH QUESTIONS**

In a high speed monorail system which runs in a closed loop rail, which was orchestrated by the floor PLC, how can we avoid during a communication failure? And when we use a secondary communication to compensate the primary communication failure, how will the secondary communication help in acting as a safety measure in a high-speed monorail system? With the sensor based safety systems in use, how will a communication is based safety is sophisticated than the current safety measure? If the communication based safety system is possible, is it possible to implement an IoT based prototype? By using the IoT based prototype for safety, Is it possible that this prototype can be implied on the current system without much change or no changes in its way of working?

### 1.3 SCOPE

The system has a primary Wi-Fi communication system which sends the position information to the floor PLC. It works in parallel with a laser based distance sensor to avoid collision. This thesis emphasis upon using IoT devices to create a secondary communication between monorail carriers and avoid collision during communication failure in any one of the carriers. The primary communication failure in the system is most likely will not happen, but the history of such incidents are also recorded in a working system. In addition to the communication based collision avoidance, hardware based safety sensors will also prevents collision. But by using IoT devices in this method will cut off the highly expensive sensors and does the same tasks just by communicating through the secondary communication.

### 1.4 LIMITATIONS AND CHALLANGES

Most of the devices needed special interface and some of the potential suitable devices cannot be used because of the wireless regulation. Zigbee devices come in different transmission frequencies, and unavailability of the ZigBee module which is suitable for the current system became a hinderence. The type of zigbee device needed for the wireless communication should have an Ethernet socket with Ethernet/IP communication option and should run in 24 volt input. which was not available at that time of designing and an another available device which is of similar type but had a drawback of using 900MHz which cannot be used in the European union [30], since it is reserved for telecommunication purpose.

The ZigBee module used in this system is a basic module and its only purpose is for prototyping. Because of that, this module don't have any special ports to connect with industrial machines. It has only one set of TTL serial port connected to the raspberry pi. To use it in an industrial system or device a TTL converter is required. Obtaining the exact industrial version of the device is not possible now. Because of the unavailability, a normal ZigBee module is used to do the task.

The communication between the raspberry pi and the PLC should have some delay to reduce the network traffic of the PLC. The PLC is communicating with multiple systems and servers to do its task. The communication with the multiple devices and servers will be done once in a few milli seconds, because of the fast process and the status of the machine to the main server and the servers commands to the machine.

During the testing an accident happened in the test platform. forced the project to be done as a simulation. A python simulation is used to show how the system will work.

## 2. BACKGROUND

In this heading, we will be looking at the background information of the applied systems and technologies. The topic of this thesis emphasize the system and technology used in this thesis. Monorail is a material-handling device with a main purpose of transporting products from one part of the factory to the next part. It consist of programmable logic controllers that controls the motors and actuators; also collect data from the sensors. Safety system used in the automation systems. In addition, the IoT (Internet of things) which are the new inclusion to the automation system, availing possibilities to access and control the machines without any need to rewrite or modify the existing PLC program.

### 2.1 MATERIAL HANDLING SYSTEM

The industrial challenges in the 18th century and 19th century paved ways to innovative methods in handling and moving of materials during transportation. During the Industrial revolution the demand for coal mining industry bring about to change its mode of transportation of coal from the mine to the processing plant. The manual labor took a lot of time and needed more space to transport multiple trolleys at the same time and needed many laborers which lead to the increased selling price of coal. To overcome all this a wooden base, leather belt conveyors are installed in mines to transfer the coal outside the mines. By this way, the conveyor systems (Figure 1 b) entered into the processing and manufacturing systems to transfer materials from one point to another point with less effort. According to the need and use, different types of conveyor belts started to evolve over time. Henry Ford changed his view on the typical belt and roller based conveyor system and modified it to overhead chain conveyors. He used the overhead chain conveyor in his assembly line to move the product to the technicians, which became a milestone in the manufacturing industry. [1]



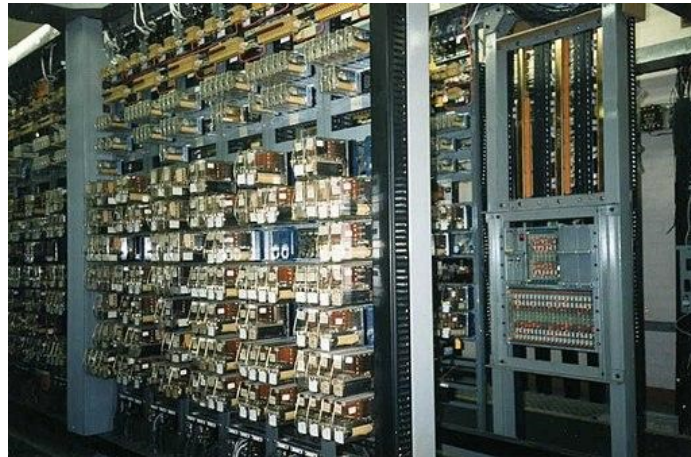
*Figure 1 a) Monorail conveyor in Ford Motor Company and b) typical Conveyor unit [19][20]*

The material handling system is the backbone of any manufacturing facility. The raw materials and the partially finished products transferred between two or more process locations after completing a process and after finishing it to the storage location. M. Mashaei and B. Lennartson [8] stating that a material handling system is a waste but necessary (auxiliary system), because it does not transform a part, but it supports adding value activities by reaching the part to operation machines. They also explained how the automated material handling system operates and transports the product within the process environment. This AMHS is integrated along with the process machineries. Some of the different forms of the AMHS are conveyors, lifts, automated guided vehicles, monorails (Figure 1 a), material-handling robots. These material handling technologies improved over time and purpose, and advancements are primarily involved in reducing time to transfer the products or raw materials and space saving or reducing dedicated space for material handling. Here in this topic, the monorail is evolved from the conveyor systems for a faster rate of transfer in a long line of multiple production units. The conveyor will be slow and consumes more space, but the monorail is overhead making the floor space available for other purposes. Also, Igor et al., in their paper [9] on comparing the typical conveyor and monorail system that “If material handling is hand operated then the potential for damage increases from 1%-3% due to the inexactness of operator handling techniques and physical limitations”. The speed difference between overhead monorail vehicles (OMV) and continuous flow transport (CFT) system is acute. Igor et Al stating, “Based on the average delivery time metric, the OMV outperformed the CFT, with delivery time values being approximately half as large as in the latter case”.

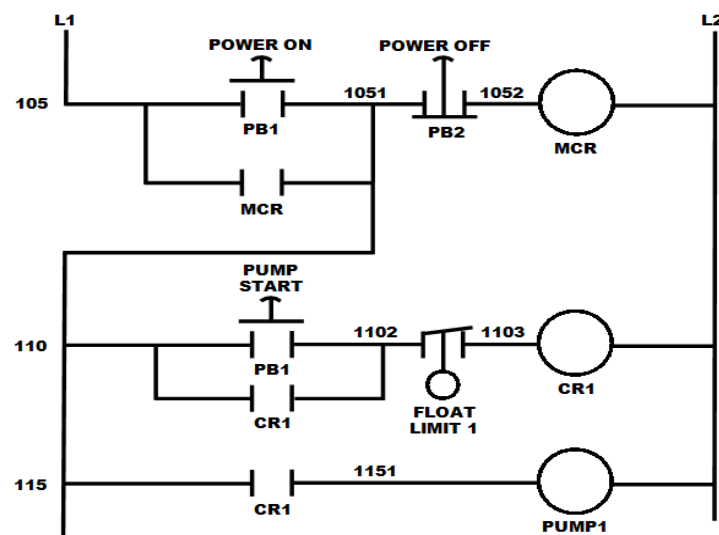
## **2.2 PROGRAMMABLE LOGICAL CONTROLLERS**

According to the Cambridge dictionary, digital means “recording or storing information as series of the numbers 1 and 0, to show that a signal is present or absent”. The digital control system is a program of a set of digital logical conditions which has to be satisfied by the state of the digital inputs to control the digital outputs. In the early days, the mechanical relays are used for creating the logic for the digital system. It is hardwired programming using physical components to obtain specific logical conditions for the system to operate. Also, one relay can control only one bit at a time; this made this hard even to make a smaller system with multiple conditions. Moreover, the relays require large space, maintaining them is hard due to its mechanical properties. The mechanical relay has multiple disadvantages such as arc formation between contacts, wear of contacts, fusing of mechanical contacts, contact failure due to spring breakage, low switching speed [28] . These disadvantages lead to usage of the highly expensive special relays. Apart from this the development of such system also faces problems such as troubleshooting the relay control system is harder, the wiring will get confused, in case of any misconnection the logic will not work and its harder to track down the error. Advancement of the semiconductor lead to the replacement of relays with transistors and the size of the transistors reduced leading to the availability small sized PLC. The size of the system is reduced and the controller became a device instead of a huge system by itself. After PLC replacing the old hardwired relay systems, it became compact and

reliable and the maintenance and troubleshooting became easier. To make the programming easier for the people who worked on the relay logic circuits, the ladder-logic programming method is designed to look close to the relay logic. The PLC became the default in automation industry because of its reliability, easy to program and mainly adapting from the old form of relay logic to the new plc control. The troubleshooting of errors is easy in PLC comparatively with the relay logic control. The PLC can be mounted inside the cabinet due to its small size.



*Figure 2 Relay logic racks for controlling [13]*



*Figure 3 Sample Ladder Program [14]*

PLC comes in different types, block PLC's or Integrated PLC's which has fixed IO ports and is used in smaller systems and fixed systems which has no way of future hardware modification and expansion. In these integrated PLC's there are multiple types according to its size like Pico PLC's, Nano PLC's, Mini PLC's, Standard PLC's. The other type is a modular PLC's which can expand by adding extra inputs or output modules, also multiple kinds of add-ons are available nowadays. These types of PLC are used in big

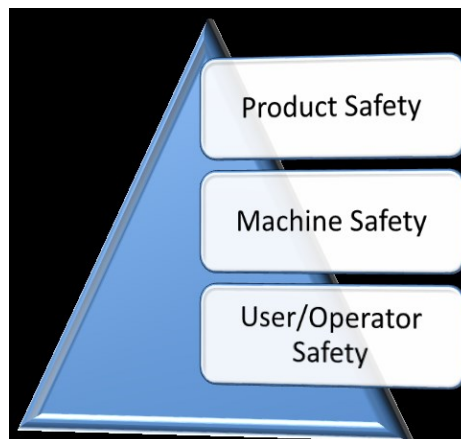
factories and processes with a lot of inputs and outputs. The Soft PLC is a virtual PLC controller, which runs from a modern computer and does the same as the normal hardware PLC. It is no more than a computer program, which emulate the functionality of a traditional PLC and uses the processor of a PC for the computation of the logic. The soft PLC's can run in the normal commercial computers available but communicates with a control module to connect to control the devices to control.

The brain of the material handling system is a PLC. It does all the calculation and Boolean logics and functions and controls the IO port.

## 2.3 SAFETY SYSTEMS

In an automation system, the safety plays an important role by avoiding any accidents from happening during the process. Some of the safety systems stop the entire process; some safety systems disable a part of the system or specific machine. Using proper safety systems, the accidents can be avoided from happening to prevent injury to operators and factory personnel and avoids damage to the machinery and the products. The safety priority is as follows

- i. User/Operator Safety
- ii. Machine safety
- iii. Product safety



***Figure 4 Safety priority hierarchy***

The user or operator's safety is considered as the priority in the safety hierarchy it is the very basic of the safety system. The user or operator should not get hurt by any means when operating the machine. The next is the machine safety, which is the second most priority where the design should not allow any possibility of damage to the machine while in use. The product safety and machine safety overlap each other in different situations. In some special cases the priority gets swapped between them according to what kind of

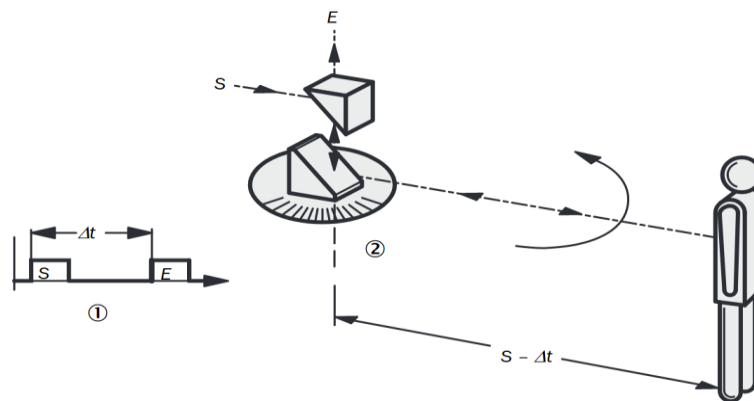


product they are handling and what effects it could cause if got damaged. For example, in a toxic chemical handling unit, the product safety is important than its handling machine, since chemical leak can harm the operator or any other in the vicinity.

The safety system gets information from the surroundings and react to it. Sometimes the safety systems are triggered through the logic of the control program. In an automated mobile unit, a distance measurement system will be used to measure the distance from other objects with the moving system. Some of the moving system which uses motion techniques like line following, track based, laser guided, path generation algorithm. These systems use safety sensors, which are used for safety purposes. An excellent example of such system is PLS proximity scanner that uses optical distance measurement technique.

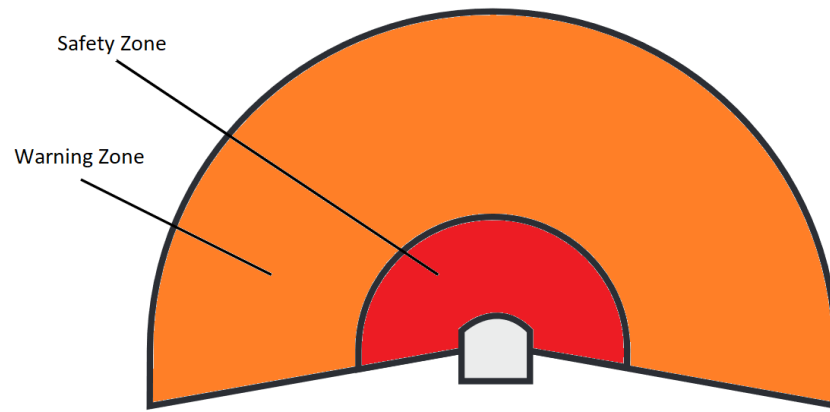
In a traditional system a PLS proximity laser scanner is used for the safety of most of the automated systems or machinery, the proximity laser scanner scans an area for any interruption and avoids any collision. The PLS proximity sensor consists of a laser diode, a laser detector an angled mirror with a rotating mechanism which rotates the mirror. The laser is flashed on the angled mirror, and the detector will detect any reflections. The laser is fixed, and so is the detector. The degree of rotation of the mirror is measured, and for each angle, the laser scans for any obstruction. The scanning is done in a specific height covering the horizontal axis around it, the firing of the laser can be controlled only for a specific angle or just a few points, according to the wish of the user.

The figure shows the basic structure of the scanning mechanism, S is the source of the laser, E is the detector which detects the reflection.  $\Delta t$  is the time taken for the laser to start from the laser diode and get reflected and reaching back to the detector.



**Figure 5 Working of a PLS proximity sensor [15]**

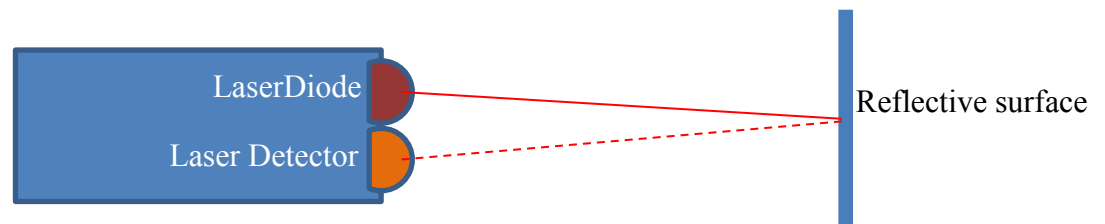
The scanning is called a light screening, if there is any obstruction in the light screen the scanner will determine the distance of that specific obstruction by the reflected rays. This screen is having multiple levels also known as fields or zones, a few warning zones, and a protection zone.



***Figure 6 Different zones of a PLS proximity sensor [15]***

This method is the traditional way of protecting the autonomous vehicle. In the monorail, it must find out whether there is another carrier in its path to avoid the collision. The floor control unit takes care of the collision avoidance by having a collision avoidance mechanism, which gets the position information of all the carriers, slows them, and stops them from colliding with each other.

In some of the systems like the monorail carrier high precision position encoders are used to track them. Since they are in a controlled environment and gives the exact location the collision prevention is calculated mathematically by calculating the distance between two or more devices and a laser distance measurement sensor is fixed on the front of the carrier that it will measure the distance between itself and the carrier in front of it. The laser distance sensor is aligned to a reflective surface to get optimum measurement.



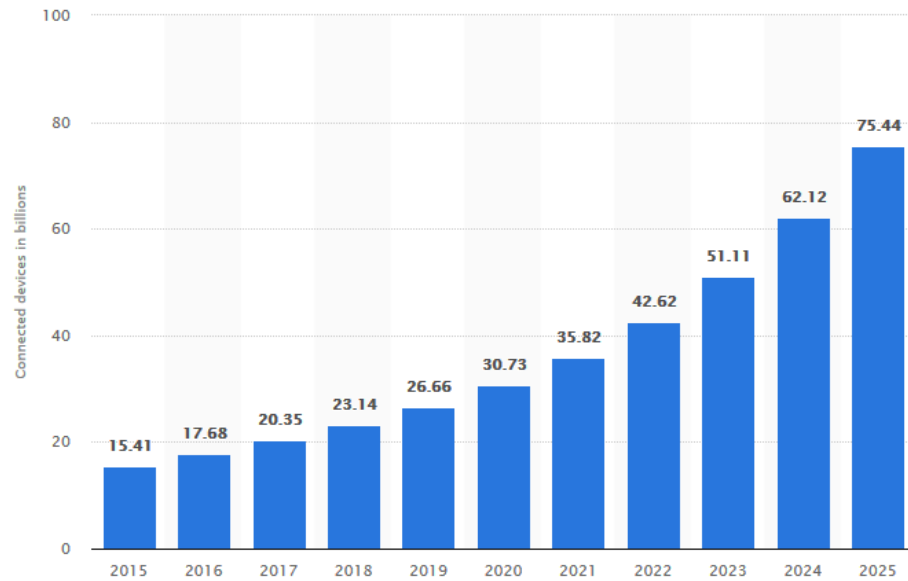
***Figure 7 Laser Distance measurement sensor***

## 2.4 IoT DEVICES

In the development of the internet and integration of the devices in everyday life making everything accessible by the touch of the screen. The development of the cyber-physical system is growing fast, and companies estimate the number of connected devices and is increasing exponentially [16].

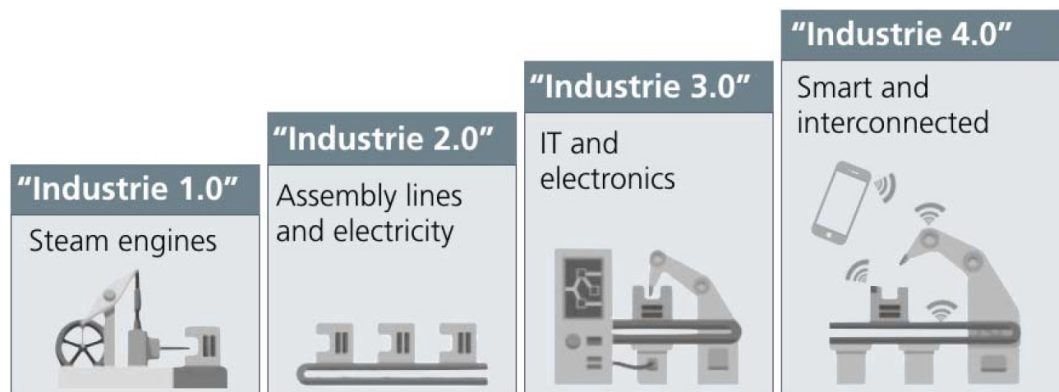
The IoT industry started as soon as the very first network is created, it is the communication of two devices between each other to do a specific task. That is why even the very first computers used inside a network can be considered as IoT devices. However, it has been mostly considered as physical devices which communicate between one or more devices using the internet as the communication platform thus by making it controllable through the network. The very first internet controlled device is varying from article to article, but when comparing the available data and sorting it by year gives us the first IoT device is a toaster built by John Romkey and Simon Hackett in 1990 which they demonstrated it in the 1990 Interop which becomes a hit of the event [17]. Ron Schneiderman in his article “Internet of Things/M2M A (Standards) Work in Progress” states that “Kevin Ashton, the co-founder of the Auto-ID Center at the Massachusetts Institute of Technology (MIT) is usually credited with coming up with the term IoT in 1998” [7]. Even though the IoT technology was not that much common among the engineers in its beginning stages, there are many other normal computer-controlled devices which have been used by enthusiasts to control various devices using a computer. During the early stages of computer-controlled devices which are built by the enthusiast were by sending messages through the serial ports. In some cases, the parallel ports were used where few of the parallel port pins can be used as Boolean inputs to the custom electronic circuit boards. Which will send the Boolean value of the pins to the logic where it triggers a relay. Then the development of the HTML and web-based programming languages made it possible for many to try this new way of communicating with custom devices and control them through the internet [18].

This paved the way for the fast development of the modern IoT devices. Also, the availability of microcontroller development platforms like Arduino and other kinds of microcontrollers boosted the IoT development. Many enthusiasts-built devices controlling all kind of actuators and accepting all kind of sensors. Then started the development of add-on cards for those microcontroller development boards in which communication boards became a vital part by allowing the devices to communicate between each other in many ways like Bluetooth, Ethernet, wireless and many more ways possible. The upgrading of simple hobby electronics between enthusiasts from the old form of electronics to a more customizable microcontroller-based playground. The microcontroller-based controlling has made it possible for many to try new ideas and devices just by programming the logic inside a microcontroller. The microcontroller projects developing DIY community started to grow bigger that it paved the way for many small startups.



**Figure 8 Number of connected devices with forecast [21]**

The fast growing community and different kinds and variants of ready-made products available in the market makes IoT a perfect solution for trying new prototypes. Prototyping much easier with community backed and open-source libraries, which allows companies to try the new technologies without any, copyright issues and any kind of delay for patent usage permission. In the design principles of industrie 4.0 scenarios, mario et al., states that “The increasing integration of the internet value chain has built the foundation for the next industrial revolution called industrie 4.0. [10]



**Figure 9 Revolutions of Industrial development [5]**

The industrie 4.0 as defined by GTAI (German Trade and Invest) is the prospective fourth industrial revolution on its way to an internet of things, data and services. The internet of things collects and process data in various stages and communicates between themselves to make a network of decentralized intelligent objects, which influence the process to get efficient and reduce time of production. The sensing units, communication units are modularized that it just have to be put on top of the other module or with the micro controller. The modular structure of these IoT devices reduces time spent on hardware designing and building. Also, these IoT devices are having a decent speed that they can

be used for controlling systems which does not require high speed controls, like conveyors. Mario Herman et al., in their paper design principles for industrie 4.0 scenarios names three key components of industrie 4.0: the IoT, cyber physical system and smart factories [10].

## **2.5 WIRELESS COMMUNICATION IN AUTOMATION**

When the communication between systems through copper cables was good for fixed systems, it faced issues with systems which are mobile. And losses are more when it comes to long distance communication. As well as the longer as it goes the more expensive it becomes with the wiring, boosters, cable protecting shields and powering the boosters. But after the invention of radio communication it became so flexible to have multiple systems communicate with each other without physical connection between them and it paved way for the modern mobile systems to work effectively.

The wireless communication technology grew so vast that it must be organized and standardized. All the wireless methods are standardized under IEEE 802 and has separated them according to their range of coverage it's been separated as Personal area network (PAN) as IEEE 802.15 , Local area network (LAN) 802.11 , metropolitan area network (MAN) 802.16 , wide area network (WAN) 802.20. The PAN Personal area network is a low powered network of devices communicating between themselves within a short range. The Bluetooth is a PAN technology, which connects to other devices in a short distance. The LAN Local Area Network is used for a specific area which is having longer range than PAN's and is used commonly everywhere nowadays, one of the most common and popular is the Ethernet access and its wireless counterpart WIFI which is used for connecting to the internet and other devices in the network like a wireless printer, P2P file sharing, wireless IP camera etc., WAN wide area network is a network of bigger size which will be get connected to multiple LAN's form places far away from each other. For example the internet is considered as a WAN since it is connecting the LAN's with another LAN's which are far away from them. The satellite communication is considered as a WAN. MAN is a network connection between nodes in different places in a region, which are having longer distances that LAN won't be able to reach such distances and it is primarily like a collection of LAN's connected as nodes and these nodes are connected within themselves through wired or by radio. WiMAX is an example of wireless MAN, also it is better for privacy and avoiding the WAN since it is connected to lots of other networks, which makes them vulnerable to some sensitive data. The traffic usage from the carrier can be avoided by using a MAN since this kind of MAN's were only used by government and other organizations and only between their nodes makes it dedicated to specific purposes and keeping it under control within the organization. And Ad hoc networks which are custom networks which are designed for specific purpose and are decentralized mesh network.

### 2.5.1 DIFFERENT TYPES OF WIRELESS NETWORKS

There are many wireless networks which are short ranged networks which can be used for communication between devices, they are

**6LowPAN** is a low power IPV6 personal area network which contains multiple devices connected and has its own IP address which could be used to identify itself through the network and connect to the internet. These devices are called as nodes which could be used to anything from a sensor to a controller. The nodes form a mesh network which is a self-healing network and this network is connected to the internet through an edge router which acts as the gateway between the 6lowPAN network and the internet. Since they are using IPV6 the size of the network can be huge, but the data rate is 250 kbps and the data packets are smaller in size.

**Zwave** is a low powered RF based communication technology targeting home automation with controlling, monitoring and status reading of home appliances. It operated in Sub-GHz frequency hence won't be disturbed by Wi-Fi or Bluetooth, with a data rate of 100 kbps. Its interoperable characteristics makes it easy to use the product regardless of manufacturer.

**Bluetooth** is well known for its usage in mobile devices especially for audio streaming and file sharing purpose. The Bluetooth technology uses frequency hopping technique for the communication and every packet of data will be transmitted in a specific frequency. It runs between the frequencies 2.40 to 2.483. The hopping is done between these frequencies. It can hop between frequencies up to 1600 times in a second. This frequency hopping technique helps the Bluetooth from avoiding interferences from other sources. The network formed by the Bluetooth is called piconets. The data rate of the Bluetooth technology varies according to the version of it where the Bluetooth v1 had a data rate of 700 kbps and v2 has 2.1 Mbps and v3 and v4 has 24 mbps of data rate .

**WirelessHART** stands for wireless Highway Addressable Remote Transducer Protocol which is like Bluetooth which uses frequency hopping technique but also it forms a mesh network and works in the license free 2.4 GHz frequency. The communication is coordinated with TDMA (Time Division Multiple Access) to synchronize the participants of the network. The wireless HART is composed of three main parts field device, gateway and a network manager. The field device is the remote device which is connected to the plant or process, the gateway is the link between the network and other networks or to the internet. It has 128 AES encryption facility available in it, providing data security with a unique encryption key for every message it sends. The wireless HART has a data rate of 250 kbps.

**LoRaWAN** stands for Long Range WAN is a low power wide area network targeted for battery operated remote devices in local, Regional, national and global coverage. It provides a secure bi-directional communication with long range mobility and localization services. Its optimized for low power with a large network with million devices connected. The LoRa devices are connected to the internet by means of a gateway and

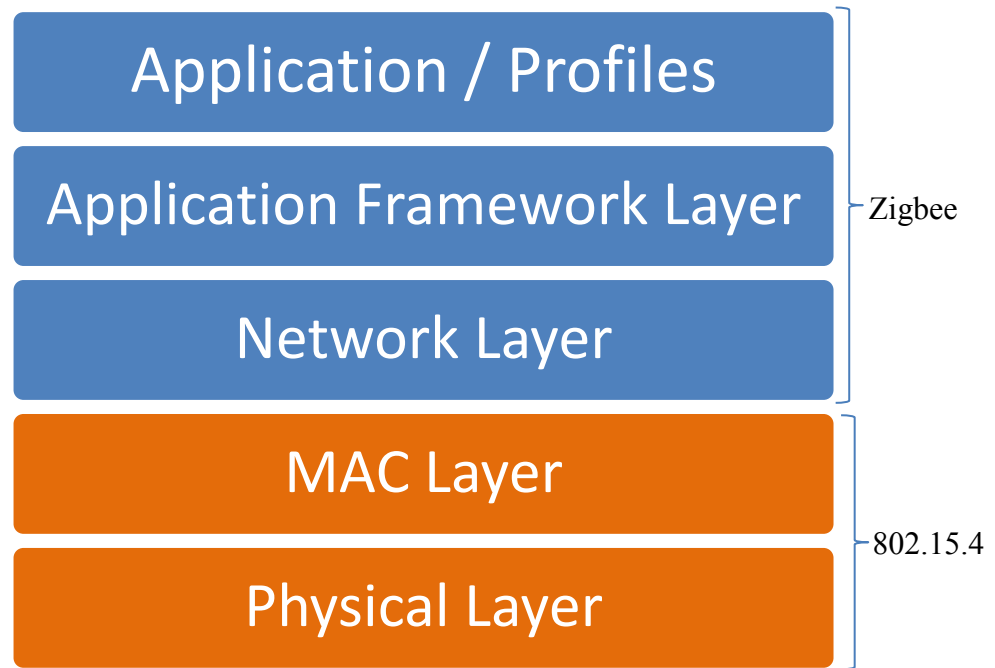
the gateway acts just as a bridge between the internet and the network making it easily available to the remote devices. The data rate of LoRaWAN is between 0.3 kbps and 50 kbps.

**WIFI** aka Wireless Fidelity is the most common wireless network which is been used by almost all the modern commercial communication devices used by the common people. It is known for its speed and is almost serves in both domestic and industrial needs. The Wi-Fi changed the old wired communication into a hassle free wireless communication between devices everywhere. But still it has its limitations.

**ZigBee** is a wireless network with short data rate, low power consuming, and mesh network supported wireless technology. It's been used vastly in home automation where device to device communication occurs. It's also used in remote sensors, the mesh network makes it special by maintaining the network independent from the control of one specific device in the network. It is also using open standards making it license free. It is well known for short range machine to machine (M2M) communication.

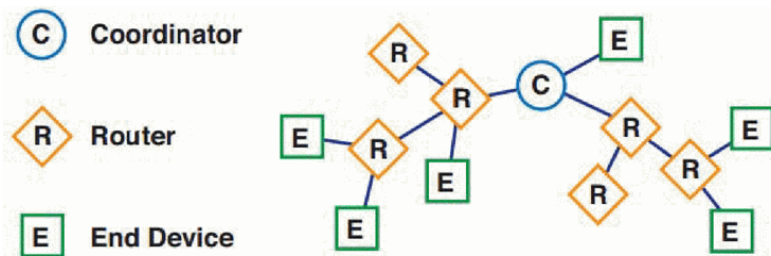
## **2.5.2 INTRODUCTION TO ZIGBEE**

The ZigBee is built based on the IEEE 802.15.4 standard with an extra layer in it to support mesh networking. The IEEE 802.15.4 contains mac and physical layer and above these another layer is created and that is the networking layer, the application layers are built on top of it. The physical layer states the physical operations such as Data rate, output power, channels, modulation, and sensitivity of the receiving antenna. MAC layer is used for point to point communication by directing the address straight to a MAC address. But it also has services like acknowledgement management and collision avoidance between data. The network layer will route the data packets between multiple devices, also plays an important role in creating a path for data hops between devices. Application framework layer contains the identifying information of a device such as profile configuration details, clusters and endpoint data. There is another layer which is the ZigBee device objects which provides features like service discovery and other advanced network management options.



**Figure 10 Architecture of Zigbee**

There are many variations in ZigBee devices. According to their frequency and purpose of the device. The ZigBee with 2.4 GHz is a common form of ZigBee since it is an unlicensed bandwidth, there are also a few bandwidths with different transmission frequency, but they might be regulated and they varies from place to place according to the zonal or regional regulations. There are mainly three different configurations in ZigBee along with a few proprietary devices, which have different type of configuration.



**Figure 11 Configuration and connections between Zigbee devices. [22]**

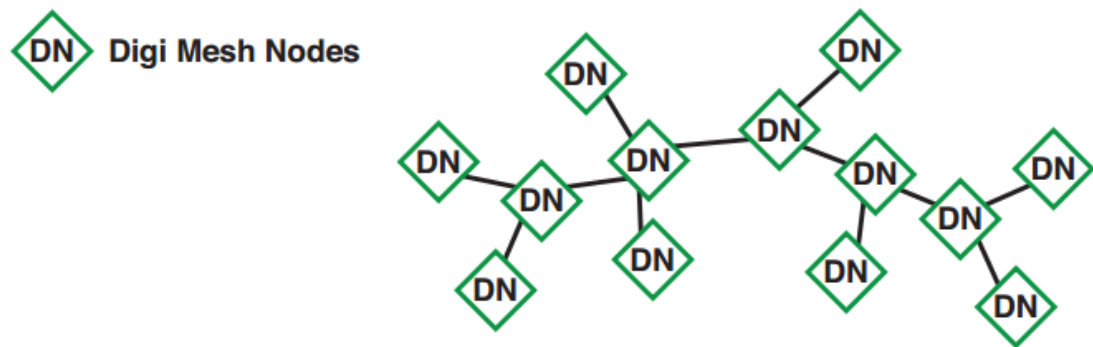
The coordinator is responsible for organizing and creating the network, whenever a new device is configured and about to join the network coordinator is the one which will register it in the network. A coordinator is a must in the network, and only one coordinator will be good because if there were two coordinators then they both will try to provide a different register value which will confuses the nodes to decide to where they want to share the data and it will collapse. Other than organizing the network the coordinator just works as a router when it is not organizing the network.



The router will discover networks and join the network to which it is configured to connect. It can also allow other devices in the same network to connect with it. Also it can allow new devices to join the network. Routing of data packets and communicating with other devices are the main task of these routers.

The end devices are the last point in the network they can discover and connect to a network but cannot route or allow other devices to connect to the network, because of its low power consumption and it will mainly runs on battery.

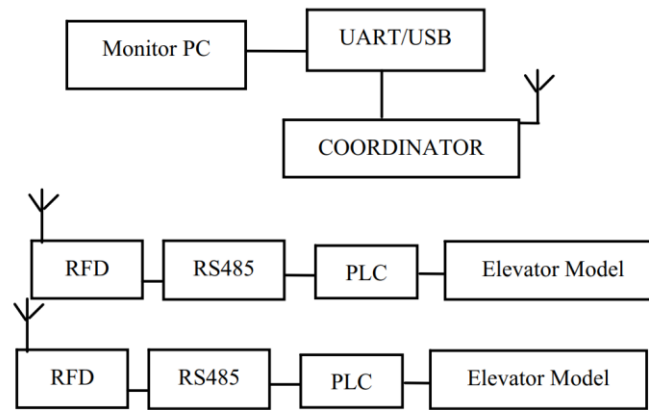
In some other variations like the proprietary digi-mesh by digi is having only one configuration which is Digi mesh configuration, it doesn't have any coordinator or router or end device. If we want to have an end device then by just changing the power options we can set the device as power saving device.



*Figure 12 Digi Mesh Network [22]*

## 2.6 SIMILAR WORKS

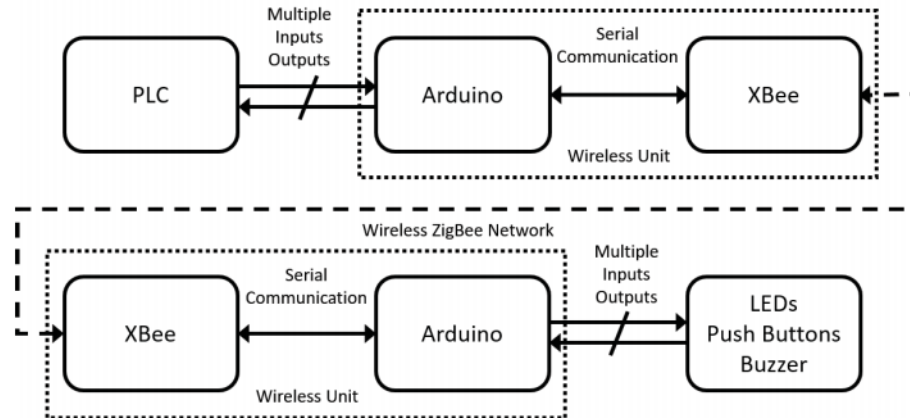
Some of the previous works resembling the thesis are considered and discussed. A ZigBee based remote control using zigbee as the communication medium is proposed by li pengfei and lijaikun [24]. It is stated that a computer will be communicating with the PLC through Zigbee modules. A zigbee network is formed using star topology with the coordinator is connected to the computer and the Reduced functional device (RFD) which can also be called as end device is connected to the PLC. The coordinator created the network in which the RFD joins and becomes one of the several nodes which will be controlled by the computer.



**Figure 13 System frame [24]**

The high performance, low power 8051 microcontroller handled CC2430 zigbee module is used for the communication with 2.4 ghz IEEE 802.15.4 complaint. The CC2430 is a complete system on chip module, which is targeted for low power wide supply voltage with AES security. With the computer the zigbee module is connected using a UART/USB converter the USB is connect to the computer and the TTL serial communication ports are cross connected to the TX and RX of the zigbee module. On the other hand, a conversion circuit is used to connect with the RS485 port of the PLC. A max3485 IC based TTL serial to RX485 is used to convert the serial data to the PLC serial input. The modules must be powered with 3.3v, which cannot be obtained straight from the PC or from the PLC. The USB to UART converter has a regulated power out of 3.3 volt to power up the module connected to the PC and a separate power supply module is used to power up the wireless module connected to the PLC. The AMS1117 chip is used for the wireless module connected to the PLC. In the software side a user interface is created using LabVIEW which will send commands to the serial port which is configured to the port where the USB to UART converter is connected. The commands from the LabVIEW are transmitted to the PLC as serial data. The PLC will get the serial information and parse it and executes the commands, which are sent by the user [24].

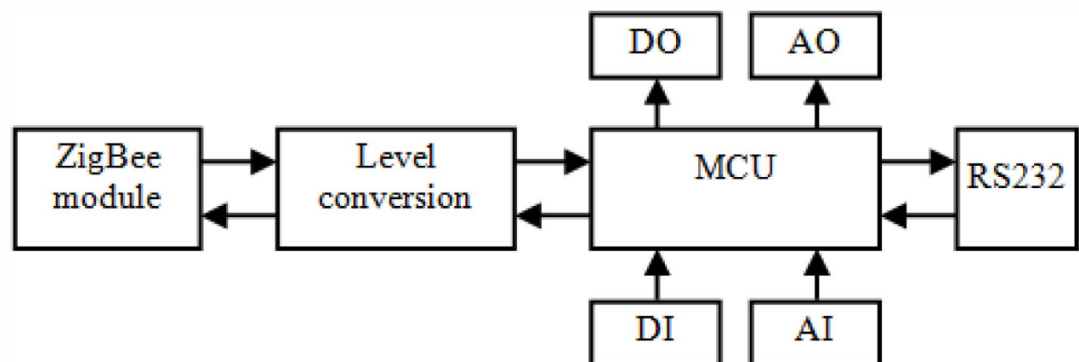
The research paper by Ali Moallim et al., [6] implementing the zigbee communication with the PLC. An Arduino microcontroller bridged the PLC and the zigbee module. The PLC inputs are connected to the Arduino output and the PLC outputs are connected to the Arduino inputs. The serial communication pins of the Arduino are connected to the Zigbee module. On the other hand, the Zigbee module's serial pins are connected to the Arduino and the Arduino OI ports are connected to the LED's, buzzers and push buttons.



**Figure 14 Proposed design block diagram.[6]**

When the PLC changes its output state the Arduino will convert the information into a serial data and transmits to another end of the communication. In the receiving end again, the data is sent to an Arduino, which will convert the serial data to trigger its output. In the same way the outputs of the sensors and switches are sent back to the PLC. In the same way multiple remote devices can be controlled wirelessly by creating a mesh network. An embedded PLC is used in this project to reduce the cost; the PLC used here is the velocio ace PLC is powered with 5V. The inputs and outputs can handle 3.3v to 24 v making it easy for the Arduino to handle it without any special interfaces. For the standard PLC types, the operating voltage must be 24 V and the input and output voltages should also be preferably 24 V. to use this kind of method an interface should have to be used to provide the required voltage levels to the PLC. With proper stepping up of voltage for the inputs and outputs for the PLC, the same implementation would be suitable for any other standard PLC's.

In one method stated in [10] similar to [6] but is using a level conversion circuit which will make it possible to use the standard 24 volt for both inputs and outputs. By connecting the micro controller based PLC unit to the ZigBee module through an opto-isolator voltage level convertor, it is possible to control the microcontroller based PLC through IO ports.



**Figure 15 System structure diagram [10]**

Figure 15 shows the system structure of the stated method with the ZigBee module, level conversion unit, micro controller unit, digital and analog inputs and outputs, and a serial communication RS232. The micro controller unit or the MCU has serial port, IO port built in together. It operates using 24 V DC power on its inputs and outputs. A ZigBee module's IO port is connected to the voltage level converter, which is an opto isolator. This opto isolator has inputs and its corresponding outputs. The inputs will be operating by low voltages such as 3.3 to 5 volts and the output side operates on 24 v. the inputs will trigger the outputs through optocouplers. Optocouplers are optical isolation components

### 3. METHODOLOGY

This chapter describes how the basic planning for building the prototype is done and how it should handle the control and communication of the PLC and wireless device. The choice of hardware and the software selected to achieve the desired result in the system are discussed here. To analyze the choice of products, the monorail transfer system's data are collected. The variables that will affect the working of the monorail carrier are checked. The requirement from the user perspective is also considered.

Some of the important parameters that are considered during the planning of the new system. The possibilities of collision, specification of the system and user preference are taken into account.

The specifications of the monorail carrier are:

1. X-axis maximum speed on rail is 5 m/s.
2. X-axis maximum acceleration is 1,8 m/s<sup>2</sup>
3. X-axis maximum movement in one direction is 300 m
4. Minimum turning curve radius for X-axis is 1 m
5. Z-axis maximum speed is 2 m/s
6. Z-axis maximum acceleration is 2,5 m/s<sup>2</sup>
7. Z-axis maximum stroke is 5 m

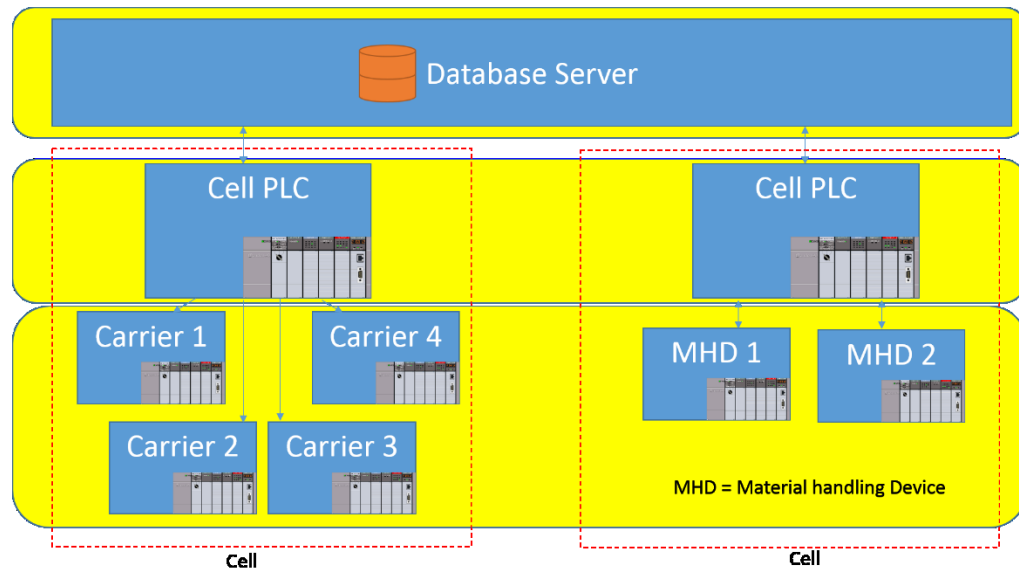
Some of the important points to be satisfied in this implementation are:

1. The communication must not be in Wi-Fi.
2. Should not have major modifications in the PLC program.
3. A delay of 200 ms is required when communicating with the PLC to reduce the network traffic in the prototype.
4. The safety clearance should be 6 meters between the carriers.
5. There are no unused inputs and outputs in the PLC, so the prototype controller have to connect only through the ethernet port.
6. Once the troubled carrier crosses the target of the neighboring carrier while moving to the maintenance position, that neighboring carrier should resume moving to its target.
7. All the wireless modules should be having the same privileges and be able to communicate with each other with ease.
8. The wireless communication system should not communicate in 5 GHz because the primary communication is working in 5GHz.

#### 3.1 ACTUAL SYSTEM

The actual system consists of a centralized control unit which is called as the floor control unit. It acts as the gateway between the warehouse control system (WCS) and the

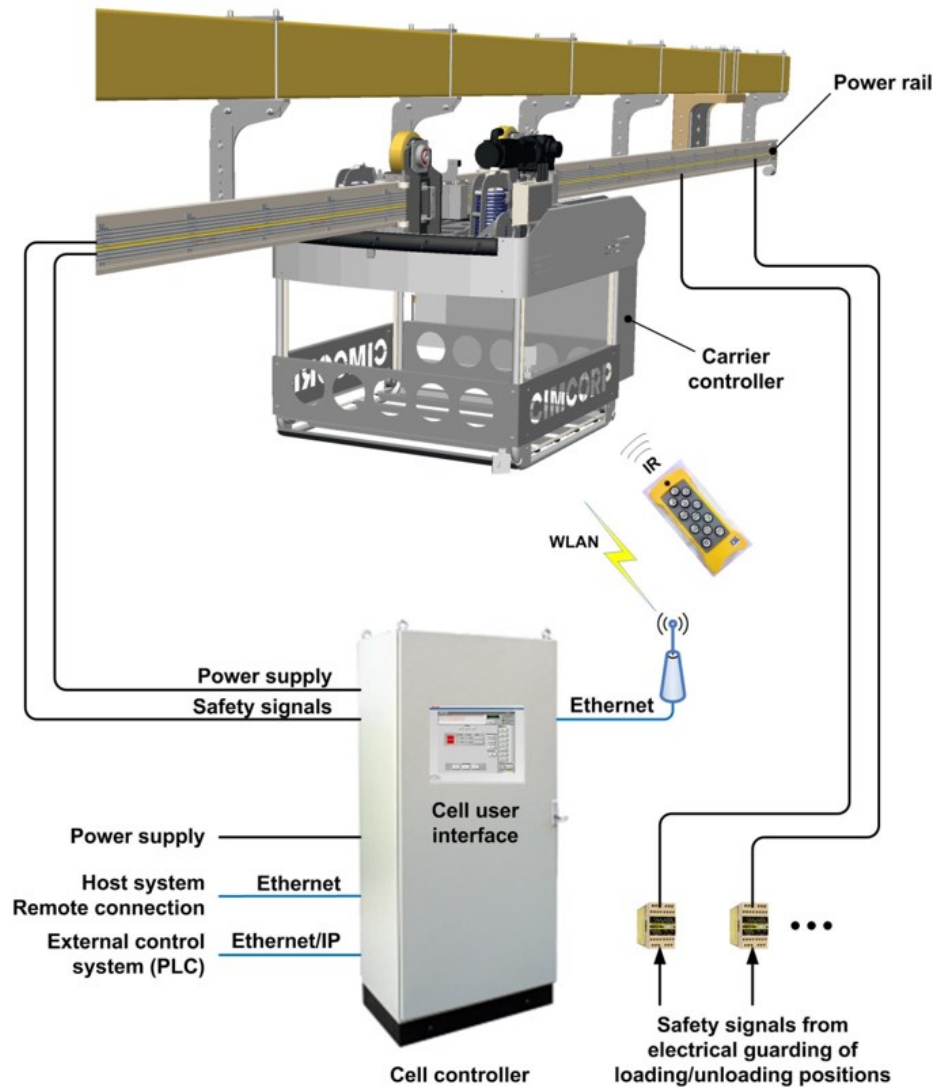
monorail carriers. The floor control unit orchestrates the carriers same as WCS orchestrating the floor PLC's.



**Figure 16 Instruction flow from higher level to lower level**

The whole system is a collection of multiple sub systems, which communicate between themselves through a communication medium to work together as one.

The Monorail system consists of three main parts, the floor controller or the cell controller, carrier, and rail as shown in Figure 17. The floor controller is the primary controller for the carriers; it will send a set of instructions to the carriers to accomplish. The carriers will complete the tasks. The three major instructions are to pick, place, move to a position. The floor controller sends these instructions in an order to accomplish it. The carrier does the ordered instructions from the floor control unit. The carrier will get the commands as a set of values, and the logic programmed in the carrier PLC, will figures out the task, and implements it. The floor PLC communicates with the monorail carriers through wireless acting as an access point with the carriers as the clients connecting to the access point. It also controls the power supply, which is given to the carriers. As the primary control and fixed the user interface is available in this floor control. It is also known as cell control.



**Figure 17 System components [29]**

The monorail carrier is an overhead rail-bound material transport system which will transports the materials from one point to another or takes it from one process to another. The transportation of the materials using monorail carriers are the fastest and more efficient in a massive production unit where the product should travel long distances.

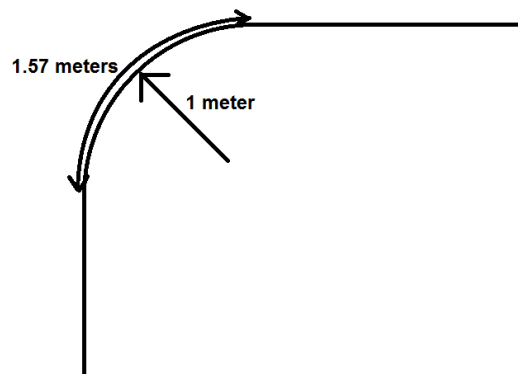
The monorail consists of a PLC, motor drive units, motors; The PLC will communicate with the floor control unit through the wireless client and controls motors through the drive units.

The carrier has a metallic body and consist of:

1. X axis Servo motor
2. Z axis motor
3. Gripper motor
4. Position sensor
5. Laser distance sensor

6. Wireless client
7. Drive units for the motors
8. PLC
9. Sensors

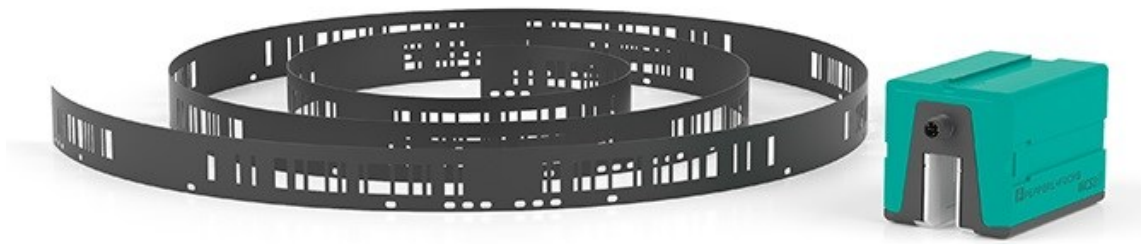
The PLC controls the motor through the drive unit according to the required task. The motors will run until it reaches the specific position from the instruction given by the primary control unit by comparing the value with the position sensor. The carrier runs on the rail at a maximum speed of 5 meters per second. The main task of the system is to pick the product from various and place it in a specific place. There will be multiple picking spots and placing spots, these picking spots are loading positions and placing positions are unloading positions. There are fixed positions for both picking and unloading the green tires. The carrier moves from its current position to the specific picking position assigned by the floor control unit. It picks the product, moves towards the assigned placing position, places, and waits for the next set of commands. The picking positions and placing positions are decided by the warehouse control system. The carrier can run both directions in the rail, but the movement is decided according to what kind of rail configuration the carrier will be used.



***Figure 18 90-degree curve of turning of the monorail carrier.***

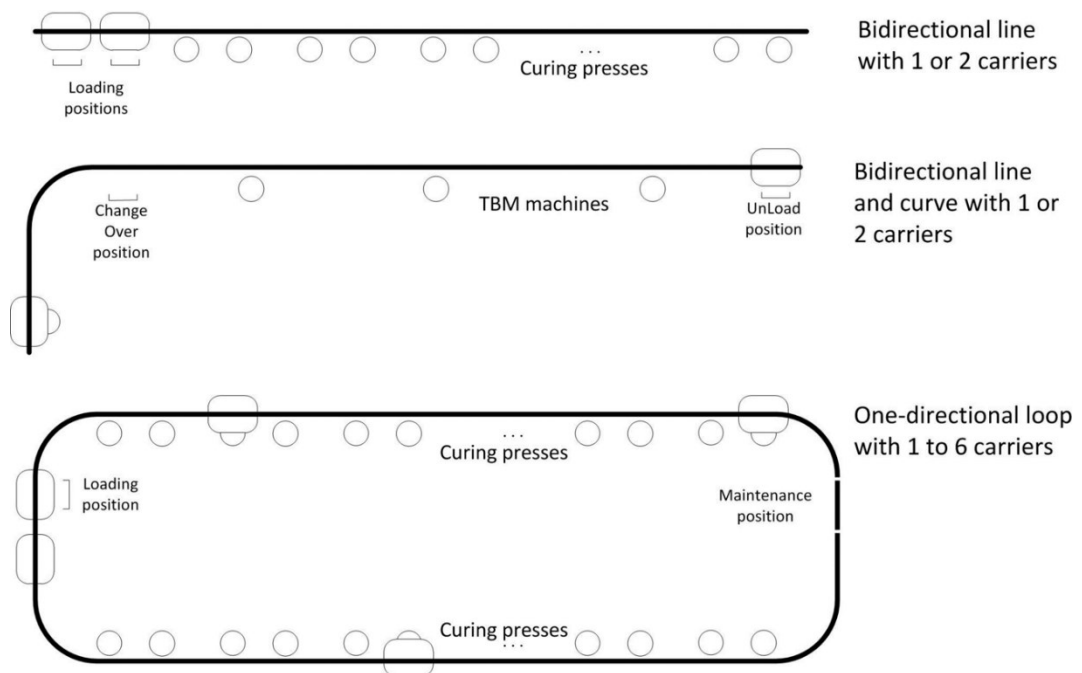
In the turning corners, the carrier will reduce its speed to avoid collision with the neighboring carrier since the distance-measuring sensor can only sense in a straight line. The rail's turning should have at least a minimum of 1-meter curve radius as specified in the machine specification (Figure 18).





**Figure 19 Position sensor with Position Strip [27]**

The rail is the platform where the carriers run. The rail is not only just a track for the carrier but also power lines and position strip attached to the rail. The Figure 19 shows the position sensor and the position strip is a non-contact type absolute positioning solution. The position sensor will give the absolute position of the carrier in the rail. The distance sensor will measure the distance between two carriers. The position strip runs along the rail providing the absolute position to the absolute position sensor. The carrier is powered by 400 VAC using the conductor bars that are fixed on the rail. The power collectors in the carrier touch the conductor bars to get power. It has five power and three safety conducting bars. These bars are insulated on all the sides with an opening on one side for the collectors to touch them. The rail is mounted on a high support made up of pillars and connected by a beam. The railing has three main configurations of transportation shown in Figure 20; they are bidirectional line, bidirectional with a single turn and unidirectional loop. The monorail carriers should handle choosing the configuration type depends on how many machines and lines. The rail configuration for this thesis is the unidirectional loop with two or more carriers in function.

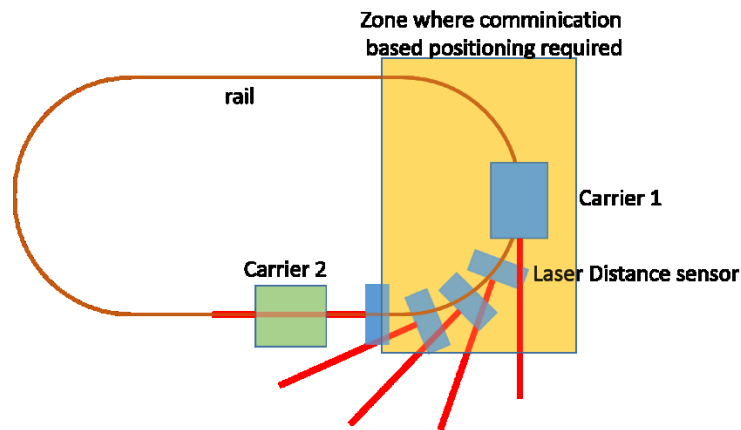


**Figure 20 Monorail Transfer, Typical systems [29]**

In the maintenance position, the rail is fixed with a lift to get the carrier up and bring down the carrier for maintenance purposes.

The collision avoidance mechanism is made of a combination of distance measurement sensor and communication-based position tracking of carriers. The distance sensor is directly connected to the IO of the PLC to measure the distance between the neighboring carrier and it. However, the communication-based position tracking is done by the floor PLC. The floor PLC will get the position of all the carriers and share the neighboring carrier position to the respective carriers.

The distance measurement sensor is a single point laser sensor that measures the distance between the laser and the reflective surface. The laser sensor is fixed on the carrier and the reflective patch is fixed on the neighboring carrier's back. When the carriers are in a straight line, the laser sensor measures the distance between the two carriers.



**Figure 21** *Curve where laser-based measurement is not helpful*

Furthermore, with the communication based position tracking of the carrier is done by the floor PLC. The floor PLC updates the position of all the carriers every program cycle. The floor PLC shares the position information to all the carriers by writing the position of the neighboring carrier in a tag every cycle.

### 3.2 SELECTION OF CONTROLLER BOARD

After considering, the above points the hardware that satisfies the above requirements are selected. A number of devices that are considered for the prototype, but only one suitable hardware is chosen among them by comparing between them. The primary need is to communicate between the PLC through ethernet port and to the wireless module through serial port. A program running in that hardware does the communication between PLC and wireless module. Since changes in the PLC program should be avoided, another controlling device will communicate with the PLC to trigger specific actions. Some of the hardware's, considered are:

1. Raspberry Pi
2. Beagle bone
3. Asus Tinker Board
4. Arduino
5. Smt32

### **3.2.1 ARDUINO AND SMT32**

They are microcontrollers, which is a complete control system on its own. They are relatively slow and communicating with the PLC needs Ethernet port. That need could be satisfied using an Ethernet shield. However, the communication with the PLC requires a specific protocol. Lack of enough libraries to communicate with the PLC made it not possible to use these micro controllers.

### **3.2.2 ASUS TINKER BOARD**

It is a complete system on board with a 1.8 GHz microprocessor, 2 GB of memory, SD card slot for storage, and interfaces as USB, HDMI, audio, GPIO's and Ethernet port. It runs in Linux and any programming language can be used to implement the control logic. The availability of the asus tinker board is hard, and lacks in developer support.

### **3.2.3 BEAGLE BONE**

Beagle bone is same as the Asus Tinker board but it has a 1.5 GHz cortex A5 processor with 2 GB of ram along with 4 GB onboard storage and SD card slot for expansion. It also has all the interfaces such as HDMI, USB, audio, GPIO and two Ethernet ports. But it is expensive for prototyping. In addition, that kind of specification is not required for this prototyping.

### **3.2.4 RASPBERRY PI**

Raspberry pi is the most common system on chip minicomputer with 1 GHz quad core ARM processor, 1 GB of RAM, SD card slot for storage. It also has other necessary interfaces like HDMI, USB, audio port, 40 GPIO's, Bluetooth, Wi-Fi and an Ethernet port. It has a remarkable community with enough support from the users and developers. It is also inexpensive and runs its own Linux OS, which makes it possible to use any kind of programming language.

The raspberry pi is chosen for the prototyping due to its inexpensiveness, good community support and it meets the requirements for the prototype.

### 3.3 SELECTION OF WIRELESS DEVICE

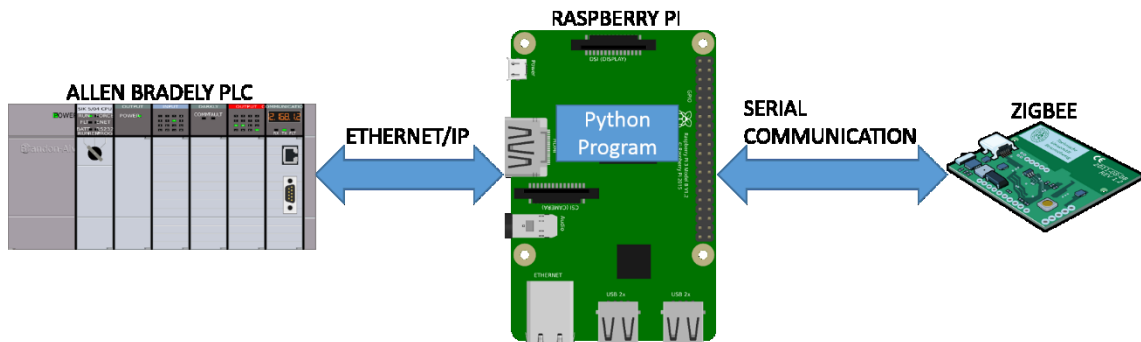
The different wireless communication technologies available and the zigbee wireless technology used here is discussed in the topic 2.5. The zigbee is selected because it is one of the commonly used wireless technology in an automation system. It has message encryption with password, good for short-range communication, and have advantages of mesh networking. Importantly it works on 2.4 GHz, it will not interfere with the primary communication, and it is license free.

After checking different kind of zigbee modules of different variants a suitable wireless module fulfilling the requirements is chosen. The requirement is the zigbee has to connect with the raspberry pi through TTL serial port. Digi is a company that sells commercial zigbee modules of different varieties. They have a wide range of products according to the range, frequency and type of interface. Sub GHz variants with 900 MHz ZigBee has license issues; they are reserved for cellular communication in Finland [30]. The xbee module zb-24 with wire antenna communicate using 2.4 GHz radio frequency suited for the prototype requirements.

### 3.4 CHOICE OF PROGRAMMING LANGUAGE

For the programming, there are three main purposes:

1. Communicate with the PLC using Ethernet/IP
2. Communicate with the zigbee module using serial communication
3. Perform the logical operations systematically.



*Figure 22 The prototype*

Python is the programming language used to write the program that establish communication between the PLC, raspberry pi and zigbee module Figure 22. To communicate with the PLC the python library CPPPO is used and to communicate with the Zigbee serial library is used. There are other languages, which can be used for the communication between raspberry pi and the PLC. Some of them are javascript with nodepccc library to communicate with PLC and using C++ with libplctag library to communicate with PLC. In addition to this, all the programming languages have serial communication libraries. For prototyping python is an excellent choice since it has less

lines of code, easy to understand and troubleshoot, which made it as the programming language for this task.

The communication with the PLC will be like query, the control board queries the PLC for the value of the tags. But on the other hand, the ZigBee module, the data shared is by serial communication. For that a message format is used. The message should contain the identity of the transmitter, the command or function the receiver must accomplish and the position of the transmitter. The three information will be send with semicolon delimited value. The delimited message format is chosen to make the message understandable, simple and short. And also parsing the message on the other end don't need any special library to parse it but by using a simple split function to get the required data.

### **3.5 REASON FOR USING SIMULATOR**

Due to an accident occurred in the test platform and making it unusable. The system has to be represented in some other way, which shows how it works. A graphical simulator was proposed to show how the prototype works with the system. The simulator has a graphical representation of the rail, two carrier, and the homing position with position control to move the carrier. This is explained in the topic 4.5 along with image.

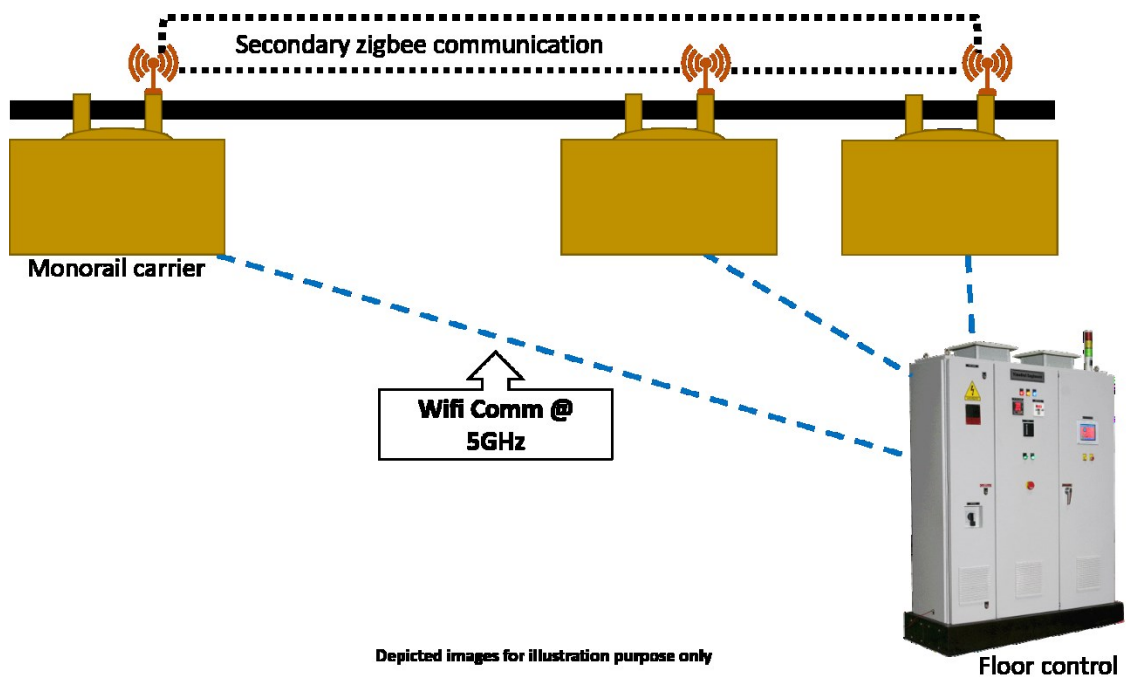
Using Tkinter library a graphical representation is created the main concept of this simulator is to simulate the behavior of the process. Since the main control program was written using python the simulator is also done using the same language.

## 4. SYSTEM DESIGN AND IMPLEMENTATION

This chapter describes the actual system which is used and the proposed modification in detail. I also describes what has been done in the implementation and the

### 4.1 IMPLEMENTATION

In the implementation, the main task was to create a secondary communication mechanism between the carriers. The carriers should communicate and share their positions.



*Figure 23 Proposed implementation*

The current system which is used should not be disturbed while the implementing the new one. Since the new system can be added to the currently available system, the changes in the working process will not be touched.

Stage 1: In this stage, the communication modules will be configured, and communication is established between two raspberry pis. These RPI's will communicate with each other similar to the communication between the monorail carriers.

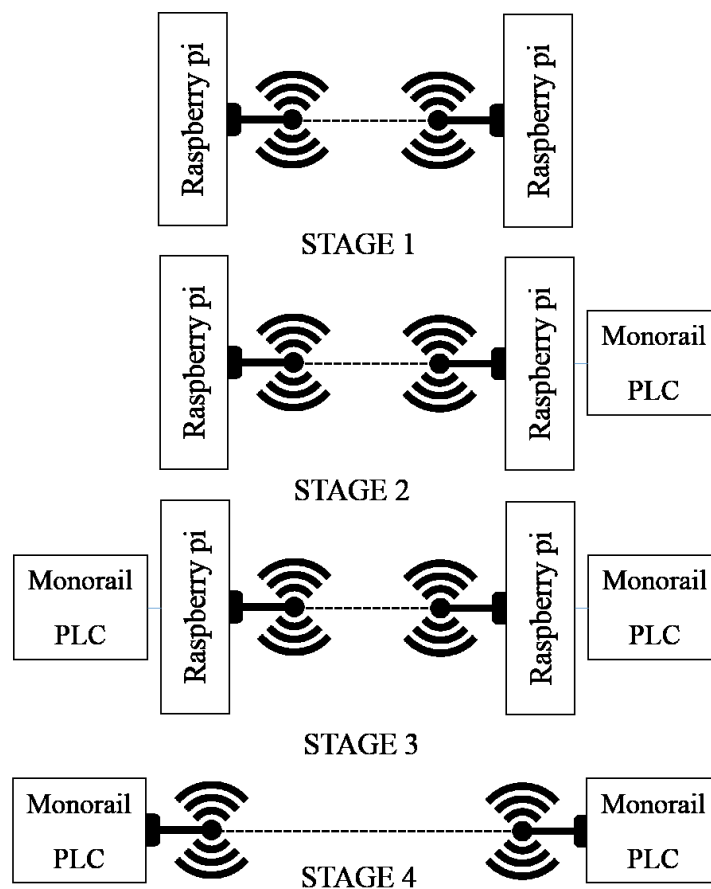
Stage 2: In this stage, one of the carriers is connected to one of the raspberry pis through ethernet. The raspberry pi will read the communication state of the carrier, and if the carrier communication is false, this will broadcast a message on the ZigBee network.

Stage 3: The monorail carriers communicate with each other using ZigBee with raspberry pi as a bridge between them and ZigBee device. The carrier is connected through an

ethernet connection to the raspberry pi, and the ZigBee is connected to the raspberry pi through the serial port. The serial connection is a two wire TTL connection between the raspberry pi and ZigBee module.

Stage 4: In this stage, the monorail carriers should be connected straight away to the ZigBee modules and communicate with each other. The ZigBee module will be connected to a serial to ethernet converter which will be connected to the PLC of the carrier. However, in this stage, the need for the change of the PLC program is required. The PLC will have the logic to communicate with each other. This stage is a future work which have to be tested and approved by CIMCORP.

In all the stages where the Raspberry Pi is used A python program is used to read and write the tags from and to the PLC and also the serial communication.



**Figure 24 Stages of implementation**

In this prototype the stages 1, 2 and 3 are executed. Since the program in the PLC should not be changed much for the prototype, the raspberry pi will be acting as the bridge by changing the values of the tags.

## 4.2 HARDWARE

### 4.2.1 ZIGBEE

Zigbee is a standardized protocol for ultra-low power wireless personal area network (WPAN) which is a joint development of a consortium of 70 plus companies together who developed this standard and promoting it. It's different from the other wireless standards because of its low price low power with more features and configurability than any other standards along with a wide range of application and its interoperability between different kinds of devices. The ZigBee is configured and the communication is tested and connected to the raspberry pi.

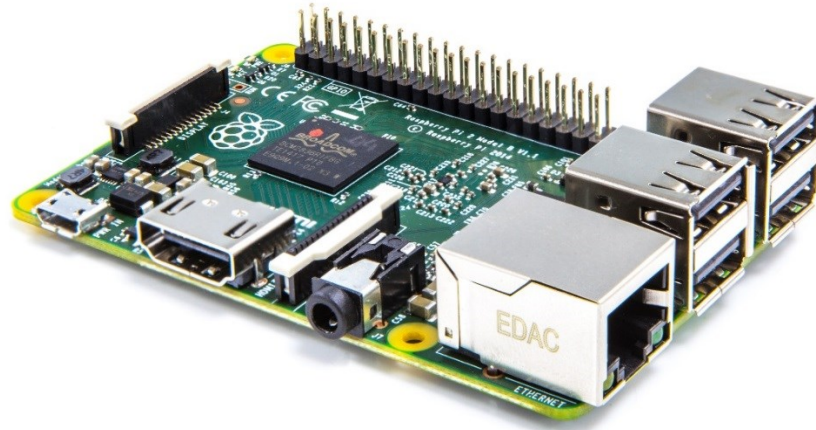


*Figure 25 Zigbee Series 2 Module. [26]*

### 4.2.2 RASPBERRY PI

Raspberry pi is an SBC (Single Board Computer) which is capable of running a linux based operating system. This SBC has 40 GPIO (General Purpose Inputs and Outputs) which has TTL serial communication ports available to connect to the serial port of the zigbee module for communication. From this minicomputer which is capable of running a python program to control the communication and read and write the PLC tag data. This minicomputer is used as the mediator between PLC and the ZigBee module.





*Figure 26 Raspberry pi [25]*

### 4.2.3 PLC

The monorail carrier operates using Allen Bradely PLC from Rockwell automation. The programming of the PLC is done by using studio 5000, which is Rockwell automation proprietary software. There are a few additions made up in the PLC program to control the carrier using the raspberry pi. Three tags are included to the PLC program, which were used for changing the mode of operation of the PLC. The three tags are Stop, Manual and Auto tags. These tags change the carrier to stop, change the operation mode to manual and move the carrier to the home position by jogging the carrier. The task is similar to the operation of a human manually operating the carrier to its maintenance position. The python program will write the value of the tag using the cpppo library.

## 4.3 SOFTWARE

### 4.3.1 THE PROGRAM

Python is used for the programming the raspberry pi which will do all the controlling and communication between the carriers. The python library cpppo is used for reading and writing the tag values in the PLC. The program will react to two main variables; one is the PLC communication status and the other is message from the neighboring carriers. If the communication between the carrier PLC and the main floor PLC is false, then it will stop the carrier and broadcast communication failure to all carriers. In addition, if it gets any message from the neighboring carriers then it will read the message and does the specific task to the specific message. If the message is communication failure, then it will check which carrier it is. If it is the carrier in the front, it will check the distance between them and react according to the distance. Here if the distance is less than 6000 mm or 6 meters then the carrier will stop and sends a message to the neighbor to move to maintenance position. Once the neighbor has reached the maintenance position then it

will send a message as it reached the position. The neighbor carrier will change to auto mode after getting the reached position message.

The communication on the zigbee side is done by message passing. A message is sent between the carriers representing their identity, a command word representing what task has to be done, and the position of the carrier. The position is shared with all the messages since the position shows where the carrier is located.

Message Structure:

*Ip Address; command ; position value\n*

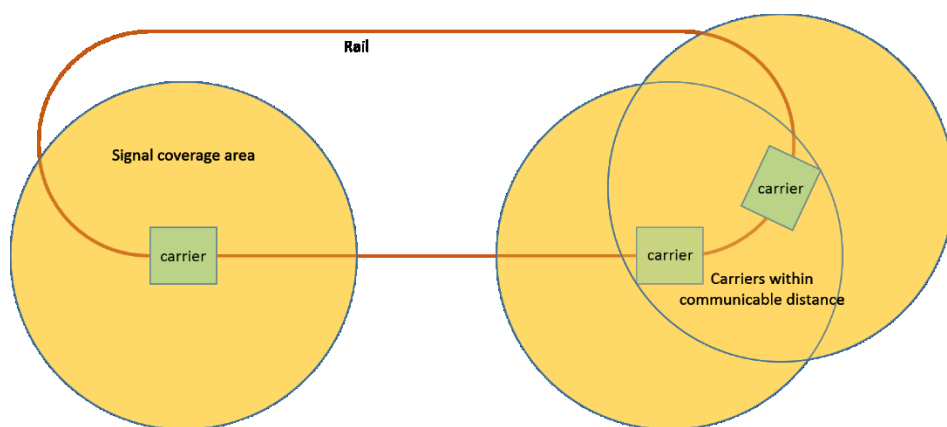
*Ex: 192.168.0.100; commfail;60000\n*

The message sent between the zigbee modules has three parts separated by semicolon. The first part is the IP address of the sender, second is the command on what task has to be done and the third is the position of the transmitting carrier.

```
output = recieved.split(";")
output[len(output)-1]=output[len(output)-1].replace("\n", "")
```

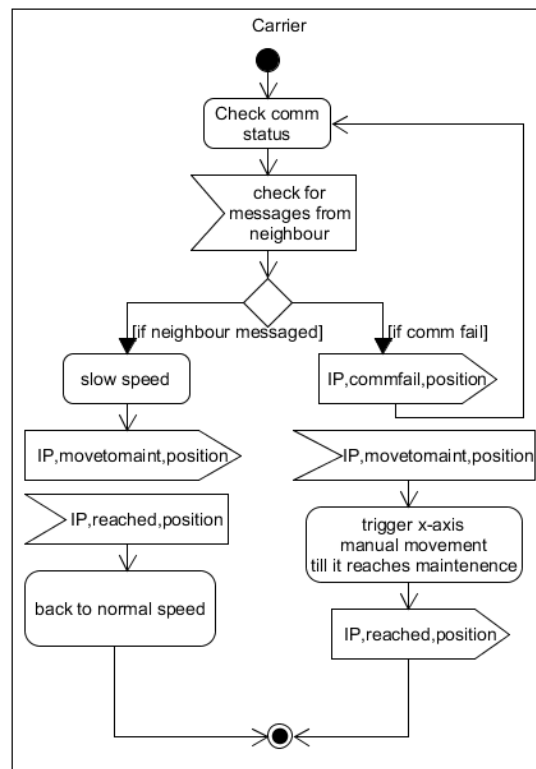
The message is parsed in python using the above two lines and the information is stored in an array for further processing.

The python program reads the tag comm\_OK value which is the communication status bit which will be true if the communication between the carrier and the floor control unit is good. If the comm\_OK tag is false, then the communication is failed, and the python program will go to the next step of broadcasting the position of the carrier through zigbee wireless network. The value of the position is obtained by reading the tag name X\_ExtEncoderPos from the PLC. It will broadcast and listen to any replies for a specific period between each transmission. When the neighboring carrier moves near to the stopped carrier it receives messages from it as soon as getting into the coverage of the broadcasting carrier as described in the Figure 27 below.



**Figure 27 Zigbee coverage and communication envelop**

And from the message it recognizes the neighboring carrier is stopped. The message contains the position of the neighbor. A reply is sent to the carrier to move to the maintenance position. The raspberry pi in the communication failed carrier triggers the stop tag of the PLC and the manual tag was written true to move it towards the maintenance position and the x jog positive tag “FromUI\_JogPosX” was triggered and it stays true till it reaches the maintenance position. Once reaching the maintenance position the PLC will be turned to stop mode by the raspberry pi. Once again, the communication status is checked and if it is true then the carrier is turned to auto by keeping the auto tag true for 10 seconds.



**Figure 28 Flowchart of the program**

The program is written in the way that it should not disturb any of the PLC program in the PLC and meant to run along with it in parallel. Only three new tags were added in the PLC program for stop, auto and manual functions.

```

def proximity(nei, own):
    ''' This is because in the loop there is a space where there is no
    position strip so position sensor gives the decimal value 419430
    length '''
    if (own == 419430):
        diff = 7000
    elif (nei > own):
        diff = nei - own
    elif (nei < own):
        diff = (maxpos-own)+nei
    if diff < 6000:
        return bool(1)
    if diff > 6000:
        return bool(0)

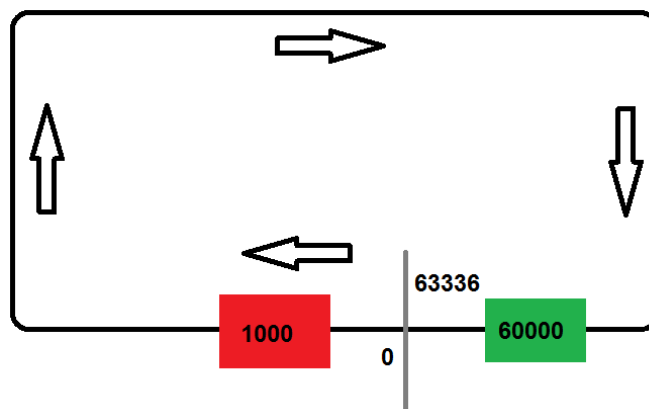
```

**Figure 29 Function to calculate the distance between careers.**

The above Figure 29 shows the function that calculates the distance between the neighboring carriers with its own position. The distance is calculated between the value obtained from the neighboring carrier by the ZigBee and the position value read from the PLC tag containing the value of the position sensor. The three possible results are

1. the neighboring carrier is less than 6 meters,
2. the neighboring carrier is not close and
3. the carrier is in home position where there is no position strip.

The no strip position is where the position sensor will give an output of 419430. That no strip position is so small, it can be neglected; but in the calculation, it gives a wrong value. By giving the difference as 7000 it can be considered that its not close to the neighboring carrier. If the distance is less than 6 meters then the carrier will stop and sends a message to the neighbor to move to maintenance position. This proximity function will compare the position of the carrier and its neighbor. If the neighbor's position is less.



**Figure 30 Calculating the distance between carriers at the end of the loop**

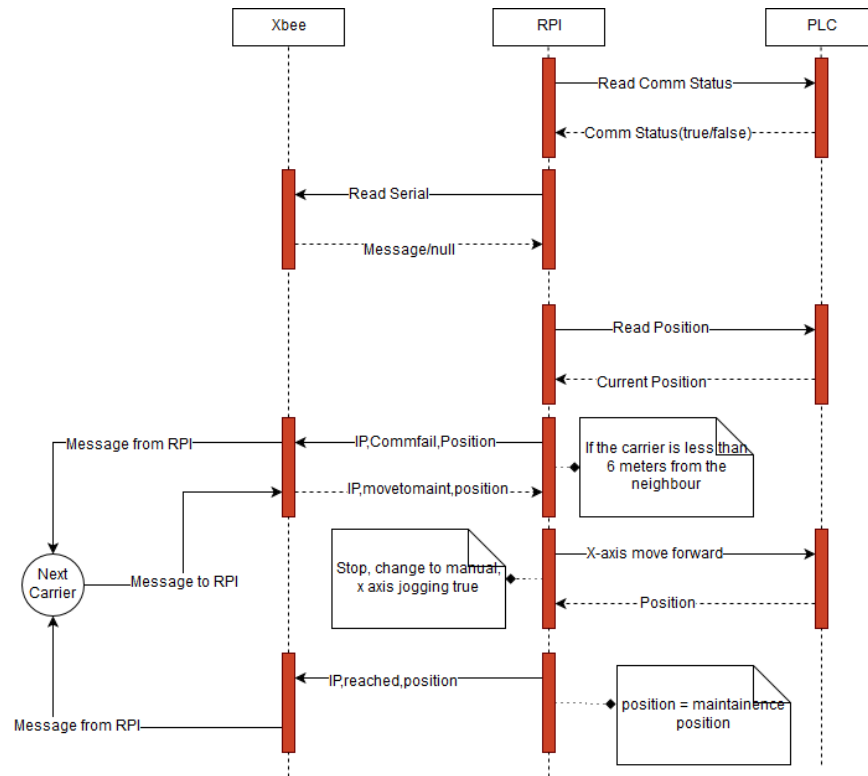
Since the rail is a one directional loop (Figure 30), the maximum position value and the minimum position value will join. At that position the neighboring carriers value will be less than the own position of the current carrier, leading to miscalculation of the distance between the carriers. A simple calculation is made to solve that issue. If the neighboring carrier's position value is lesser than its own, then using the formula below the distance is calculated.

$$\text{Difference} = (\text{maximum position of the rail} - \text{position of the current carrier}) + \text{position of the neighbouring carrier}$$

In this example shown in the Figure 30 Calculating the distance between carriers at the end of the loop the carriers position is 60000 mm and the neighbouring carriers position is 1000. The maximum position value is a constant depending on the length of the rail in mm. here it is 63336 mm.

$$\begin{aligned} \text{Difference} &= (63336 - 60000) + 1000 \\ &= 3336 + 1000 \\ &= 4336 \text{ mm} \end{aligned}$$

Once the neighbor reached the maintenance, position then it will send a message as it reached the position. The neighbor carrier will change to auto mode after getting the reached position message. The output is boolean it gives either true or false. It will be true if the distance between its own position and the neighboring carrier's position is less than 6000. If the distance is more than 6000 or if the carrier is in the no strip position the output will be false.



**Figure 31 UML diagram of the communication**

The raspberry pi will read the serial port and the communication status of the PLC. If there is no serial messages and the communication tag value is true then there is no problem with it or the neighboring carriers. If the serial message is available it will check the message. If the message is sent by a neighbor then the distance between them is calculated, if the distance between them is less than 6 meters a command to move the stopped carrier is sent back. Then the neighbor will change from auto to manual and the jogging tag will be true till it reaches the maintenance position. Once reaching the maintenance position it will send a message that reached the maintenance position.

The program is written in the way that it should not disturb any of the PLC program in the PLC and meant to run along with it in parallel. Only three new tags were added to the PLC program for stop, auto and manual functions.

The python program creates an ethernet/IP client and server communication with the Allen Bradely Control logix programmable logic controllers. It creates an UDP session to the PLC

```

def TagData(mode, tag_name):                                     # Building a function
    with client.connector(host=IP[1],
                           port=44818,
                           timeout=timeout) as conn:           # Creates a UDP connection
        operations = client.parse_operations([tag_name])
        if (mode == "wr"):
            failures, transactions = conn.process(operations=operations,
                                                    depth=2,
                                                    multi-ple=0,
                                                    fragment=False,
                                                    printing=False,
                                                    timeout=timeout) # Write Tag

        elif (mode == "rd"):
            for index, descr, op, reply, status, value in
            conn.pipeline(operations=operations,
                           depth=2):                             # Read tag
                poz = (value[0])
                if value is None:
                    print("None returned while reading %s from PLC %s " %
                          (tag_name, IP[1]))
                return poz

def PushButton(btag, secs):  ''' Function to trigger a value and reset in a
                               specific interval, like a push button. '''
    TagData("wr", str(btag)+"=(BOOL) True")
    time.sleep(secs)
    TagData("wr", str(btag)+"=(BOOL) False")

def read(tag):                                                    # Function to read a tag value.
    tagout = TagData("rd", tag)
    return tagout

def wuint(btag, val):                                             # Function to write a DINT value to a tag.
    TagData("wr", str(btag)+"=(DINT) " + str(val))

def wbool(btag, state):                                           # Function to write a BOOL value to a tag.
    TagData("wr", str(btag)+"=(BOOL) " + str(state))

```

**Figure 32 Functions which communicates with the PLC**

The Tagdata functions does the writing or reading operation. It has two modes which are reading and writing, the mode of operation is represented by wr for writing and rd for reading. The output will be the value of the tag.

By using this tag data few more functions are created to simplify the reading of the tag data. The push button acts like a push button with a time delay. It turns a tag to true, delay for a specific period and changes the tag value to false. This function is used only on Boolean tags such as x axis manual movement tag, auto, stop, and manual tags.

The read function will simply read the value of any tag. And returns the value.

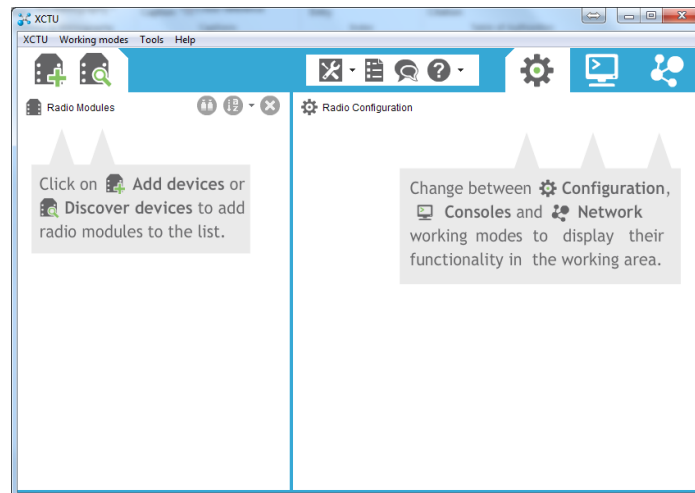
Wdint function writes an double integer value to the tag. And the wbool function will write the Boolean tag to any of the two states true or false.

### 4.3.2 CPPPO LIBRARY FOR PYTHON

CPPPO is a binary communications protocol parser library for python. By using this library, we can communicate with the PLC and read/write the tag values of any Allen Bradley PLC. The while communication is done by using an Ethernet/IP CIP control server, which will get data and write data from and to the PLC.

### 4.3.3 XCUD CONFIGURATION PLATFORM FOR ZIGBEE

XCUD is a configuration platform for the ZigBee module. The zigbee module is configured according to our need using this tool. The devices here are configured using this tool. This provides a graphical interface to change the values and upgrade the firmware for the ZigBee modules.

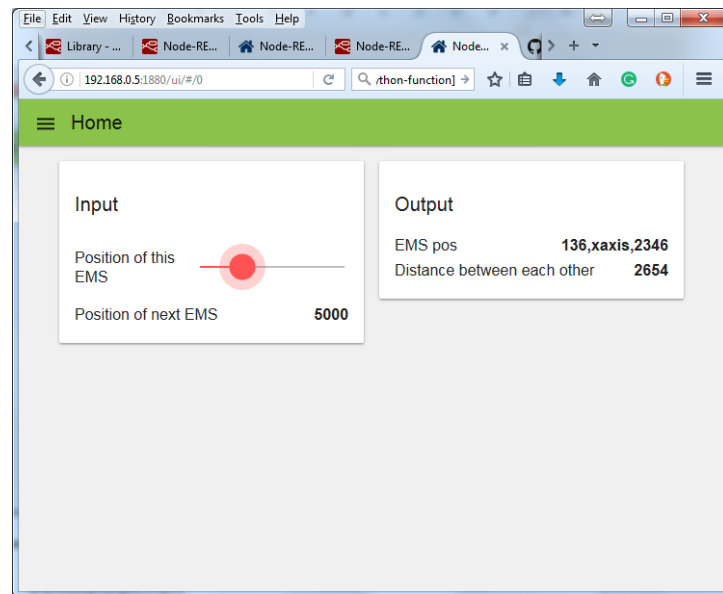


*Figure 33 XCTU configuration toolbox window*

### 4.3.4 OTHER IoT DEVELOPMENT APPLICATIONS

For primary testing and partial simulation, Node-Red has been used since it's a graphical programming language and has a good UI to view the output along with controls. The ZigBee devices are controlled by using a small interface, which will also manipulate the data from the ZigBee. The data which it gets from the wireless device will be parsed and





***Figure 34 Node-Red User Interface showing slide control and output***

Here in this UI the position of the next EMS is set to a constant as 5000 and own position is changed by using the slider control. This information is transmitted to the next EMS, there the difference between them is calculated, and the result is sent back.

## 4.4 COMMUNICATION

### 4.4.1 SETTING UP A ZIGBEE NETWORK

Setting up a ZigBee network should consist of at least a coordinator, a router. The end device is optional because it won't be making any new connections to the network.

There are a few parameters which should be modified for getting a specific kind of configuration. The ZigBee is having two modes of communication AT and API. In AT method which is a transparent method any data which will be sent to the module will be immediately transferred to the target module wirelessly. This method is much faster and application independent. All the packets given to it is just data and it transmitted it to the configured address in the firmware configuration. API is application programming language in which the parameters will be framed along with the data to be transmitted, this method is mostly used for a dynamic communication of the module with different networks and change its parameters from the data source making it dynamic used with an application which will be deciding the parameters of the transmitting signal. For example, the address and data should be framed together and should be given to the module so that it reaches the specific address. These modes can be selected by using the appropriate firmware, to change the mode the old firmware should be erased and the new one should be written in the module. Multiple firmware's are available in the library of the configuration and test utility software.

Other than the type and mode of the module there are a few parameters to be set, some of the important parameters which will be used often are

- ID - PAN ID
- DH – Destination Address High
- DL – Destination Address Low.
- NI – Node Identifier (Name for the Module)
- EE – Encryption enable
- KY – Encryption key
- BD – Baud Rate
- NB – Parity
- SB – Stop Bit

**ID - PAN ID** It is the personal network ID of the network created using Zigbee modules.

**DH – Destination Address High** First half of the unique id of the Zigbee module.

**DL – Destination Address Low.** Second half of the unique id of the Zigbee module.

**NI – Node Identifier (Name for the Module)** is the ascii name for the Zigbee module just for the user to understand easily.

**EE – Encryption enable** Enabling the encryption of data which is being transmitted.

**KY – Encryption key** is the key to decrypt the encrypted data which the Zigbee receives.

**BD – Baud Rate** is the rate of transfer of information in a communication channel.

**NB – Parity** is the bit which is used as a reference to find an erroneous packet.

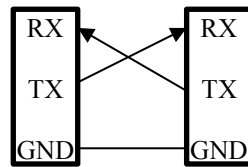
**SB – Stop Bit** the last bit of the whole transmission packet.

The xbee devices are configured in AT mode and is set to broadcast the data to all nodes of the network.

The PAN id is set as 10 for all the modules in the network.

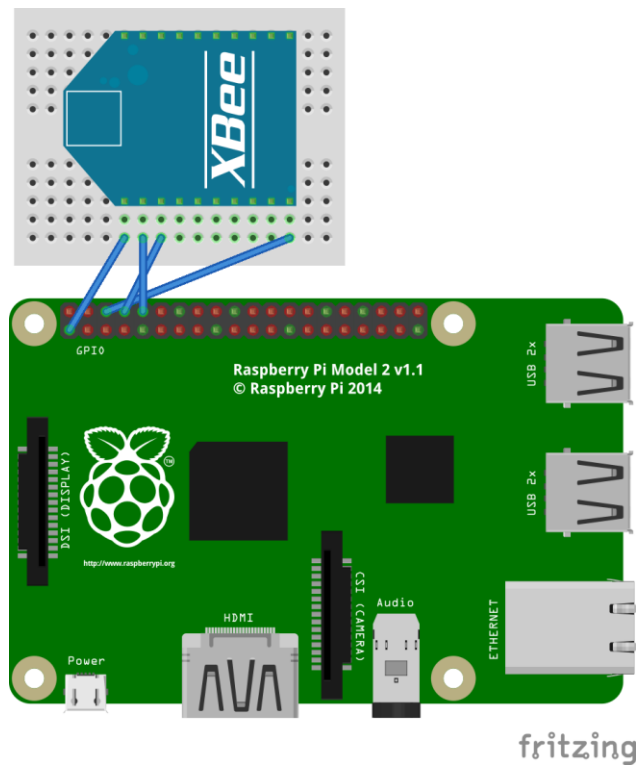
#### 4.4.2 CONNECTION BETWEEN ZIGBEE AND RASPBERRY PI

The Raspberry pi and Zigbee are communicating by serial Transistor-Transistor Logic (TTL) communication between them. The serial pins of the raspberry pi are connected to the serial pins of the zigbee. The pins 8 and 10 of the raspberry pi represents the Tx and Rx respectively are connected to the pins 2 and 3 representing the Tx and Rx respectively. The serial connections are cross connected. They are communicating at 9600 bps and the destination address high is set to 0 and the destination address low is set to ffff. By setting

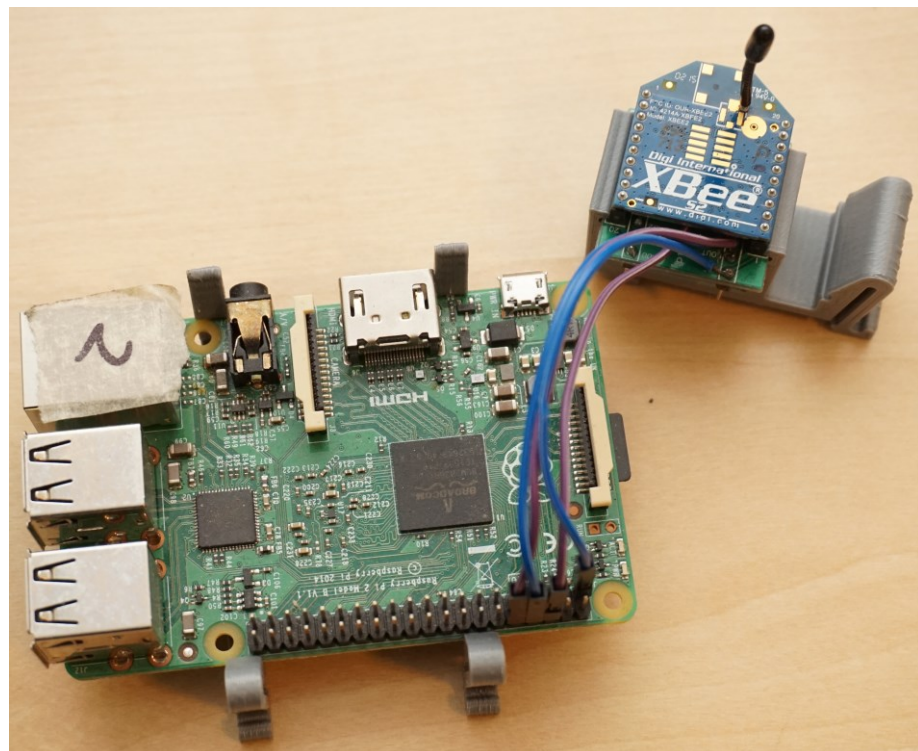


*Figure 35 Cross connection of serial pins between RPI and xbee*

the destination address to 0xffff the message will be broadcasted throughout the network with the same PAN id.



**Figure 36 Connected Xbee and Raspberry pi**



**Figure 37 The Raspberry pi Zigbee setup**

#### 4.4.3 CONNECTION BETWEEN RASPBERRY PI AND PLC

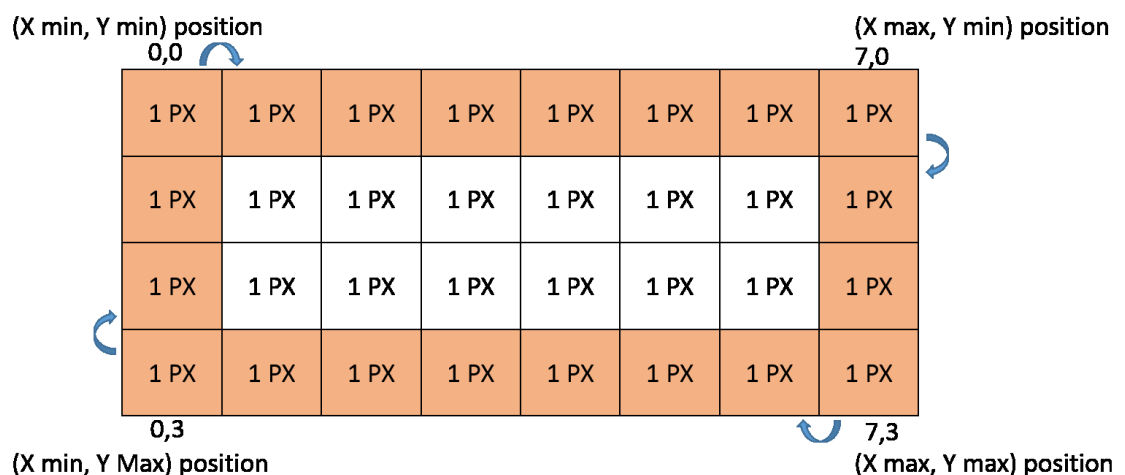
The raspberry is connected to the PLC by Ethernet. The PLC is communicating to most of the devices through Ethernet communication. The python code running in the raspberry pi will send the values to the PLC as UDP message. The python uses CPPPO library for the communication.

#### 4.4.4 CONNECTION BETWEEN ZIGBEE AND PLC

The raspberry pi will be used in the prototype only, but in the actual system the ZigBee device will be directly connected to the PLC, but the ZigBee module has a serial port for communication and to communicate with the PLC it has to be converted to Ethernet communication. For that, a serial converter is used to communicate properly. For that conversion, MOXA N5xxx series serial to Ethernet converter will be used.

### 4.5 SIMULATOR

The simulator for the system is made using python and it simulates how the system works. It has two monorail carriers one of them is white and the other is red which will stop in the middle of the path and the white will have to go to a position across it. When the white carrier starts it will moves to the target position, and when it reaches close to the red carrier it will stops because of the proximity with the red carrier, then a command is sent to the red carrier to move away from the target position and the red carrier will move away from its path to the position away from the target position of the white carrier and once it crosses to a safe point away from the target of the white carrier, the white carrier will start moving to its target, once it reaches its target the red carrier will moves towards the homing position which is also the maintenance position of the carriers.



*Figure 38 Position logic in the simulator*

In the above example Figure 38 each block represents a pixel. To convert the x,y coordinates to absolute position. By comparing maximum and minimum positions of the

canvas the absolute position is calculated. Each pixel is linked with a variable, which is an integer corresponding to that position and will be called as location. The carrier (image) will move one pixel at a time in the colored pixels. When it is in (0,0) till (6,0) the carrier will move from left to right, when it is between (7,1) to (7,3) it moves down, between (6,3) to (0,3) right to left and (0,3) to (0,1) it moves from down to up.



**Figure 39 Simulator Window.**

The grey box in the homing position and the white and red boxes are the carriers 1 and 2 respectively. For example, if the carrier 2, the red one is having trouble communicating with the main floor PLC, the white carrier will move till it reaches near the red carrier. The red carrier will start moving till it is away from the white carrier's target, and after moving away from the target, the white will reach its target. After the white reaches its target position, the red carrier will start moving towards the homing position.

The positions of the paths are taken from the pixels of the whole simulator. The pixel values are converted to positions. The proximity is calculated by comparing the positions of both carriers. If the proximity is true, then the white carrier will stop; and the red conveyor will be moved away from the white's target by moving it far from the white carrier's position. Once the white carrier reaches the target position the red carrier which is away from the white will start to move from its position to the homing position.

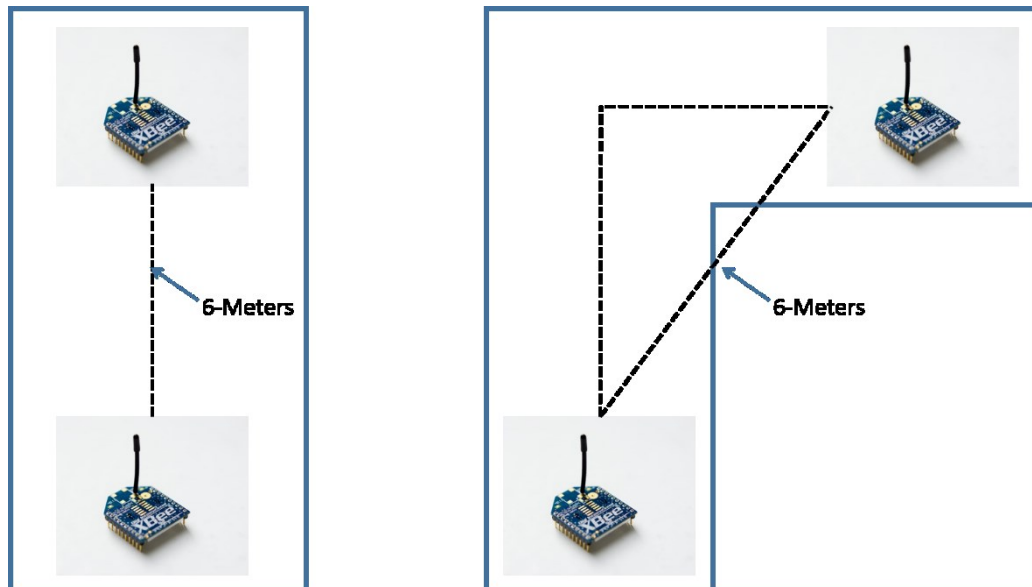
## 5. RESULTS AND DISCUSSION

### 5.1 TESTING

#### WIRELESS TESTING

The performance of the wireless connection is tested with doing some test runs with known parameters. The data transfer parameters should have to be studied to determine the performance or quality of transfer of data. To determine the quality of data, transfer the devices should run with a known distance between each other and known amount of data to be transferred. By this way the performance can be determined for a wireless communication. There are two main testing methods, they are range testing and throughput testing. In the range testing the quality of the transmission is checked by having definite distance between the wireless devices and within a specific time the transferring of as many data as possible. The packet count is compared and the number of lost, errors and received packets were calculated to determine the quality of data transfer. In throughput test with a specific time multiple packets were sent to the destination device, and the amount of the data that could be possibly sent per second is calculated.

A test setup is made by



**Figure 40 Test Setup a) Line of sight b) non line of sight**

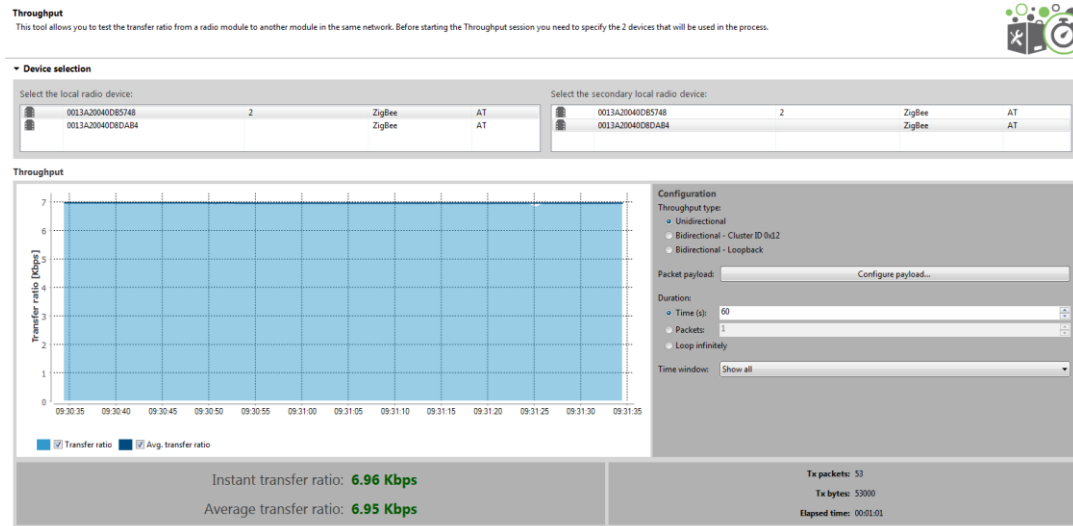
In the zigbee system, there are two different distances are used. One is 1-meter testing and 6-meter testing, since the need of transmission is 6 meters. The 6-meter reference is taken because the carrier will travel for 6 meters if it runs without break and without power due to the kinetic energy stored when running in full speed.

The testing is done in different situations and different distance. When the devices are not in sight and the devices are in sight to each other.

## THROUGHPUT TEST

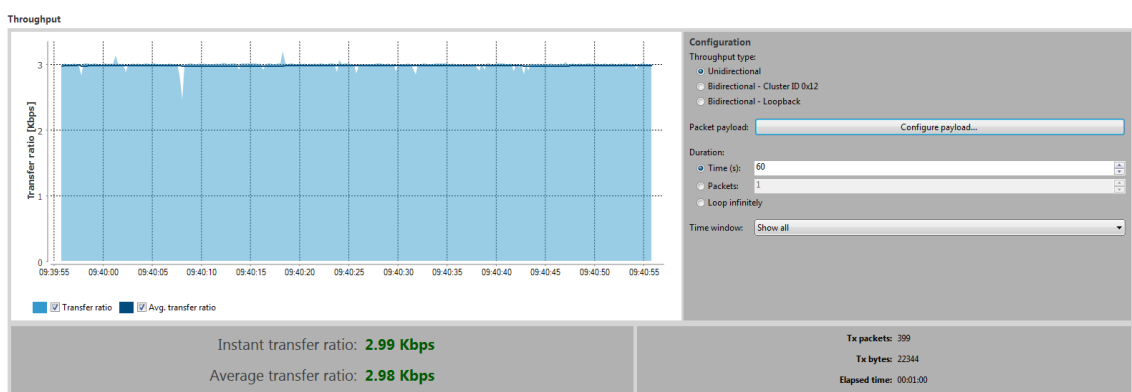
In the throughput test, the wireless devices send packets between themselves for a fixed period and calculates the amount of data transferred between them. Throughput shows the amount of maximum data can be shared between the two devices.

The throughput test for the 1-meter test showing that the average transfer ratio of 6.95



**Figure 41 Throughput test 1-meter test.**

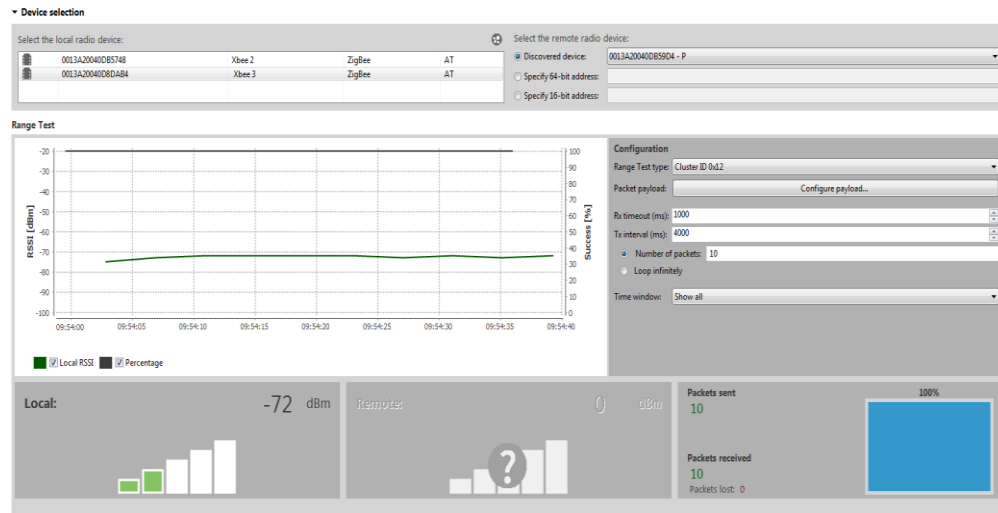
kbps during a time of 60 seconds with 1-meter distance. The devices are insight to each other. The time taken for the test is 60 seconds and the data transferred between the devices is 53000 bytes transferred. Since the devices are closer to each other the data transfer is better.



**Figure 42 Throughput test 6-meter test**

The throughput test for the 6-meter test reduced its average transfer ratio nearly half of to the 1-meter test with only 2.98 kbps with the same 60 seconds interval. The devices are insight to each other.





**Figure 43 Range Test 6-meter with obstruction in LOS**

The Range of coverage is measured by making a test by transmitting data to a device which is 6-meters away from the transmitting device and has interference between them.

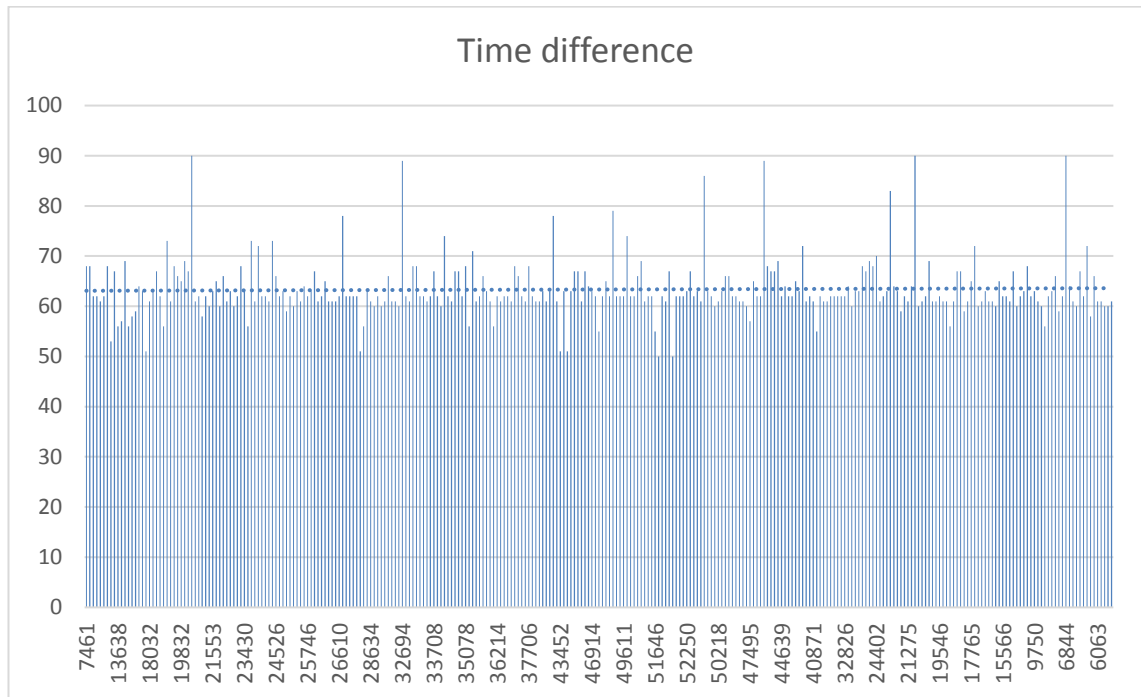
### **SPEED TESTING:**

The speed test is taken to find the time taken by the raspberry pi to read the values between two consecutive readings. The readings are taken at full speed of 4 meters / second.

A python program is used to take the readings from the raspberry pi. This test is done by using 5 steps which are:

1. Save the current time in millisecond to T1.
2. Read the tag value.
3. Save the current time in millisecond to T2.
4. Time taken to read =  $T2 - T1$
5. Save the reading in CSV file.

The average difference the carrier travelled between two consecutive readings in an average 120.22 mm with maximum of 315mm and the average time taken for the program to read the values in an average from the graph is 63 milliseconds with a maximum of 90 milliseconds. The PLC is communicating with the main server every cycle and the traffic in the network will be high. The data sent through the network are highly sensitive due to actions of the system is decided by these data which is sent to the server. The communication between the raspberry pi and the PLC is given a proper delay to reduce the load on the PLC. A delay of 200 ms is applied in the prototype, but in the future work it will be checked every cycle without any delay.



**Figure 44** Time taken for the Python program to read the position.

## 5.2 DISCUSSION

The research questions which are answered in this discussion.

*How to avoid collision in a high-speed monorail system during a communication failure?*

In general a sensor will be used to detect any obstacle on its path. The monorail has a laser based distance sensor in it which will measures the distance between itself and the neighboring carrier. In the corners the laser sensors are not much effective to sense the neighbor, so a communication based mechanism which will share the position with their neighbors is created as a software safety. In the old system the last known value of the neighbor is used to avoid collision.

*How will a secondary communication help in acting as a safety measure in a high-speed monorail system?*

A secondary communication which is independent of any other external devices or services to communicate between themselves is formed. They can communicate between each other one on one and share their position values. By sharing their position they can avoid collision by comparing the distance between them.

*How the communication based safety better than the current safety measure?*

In the corners the laser based distance sensor will not be effective because of it only travels in a straight line, also the system already uses the neighboring carrier position values through the floor PLC but it requires the primary communication. To compensate the communication break, the new system will share the position through the secondary communication making the neighbor aware of where the carrier is located hence avoiding collision. Also the sensor based safety will shows the distance between the rear of the carrier and the laser sensor. In the communication based safety the actual position from the absolute sensor is shared giving the exact location in the rail.

*Is it possible to implement an IoT based prototype?*

An IoT based prototype is implemented, but instead of connecting to the internet and depending on internet, this prototype is self-contained in its own network. Working in its own network provides safety to the system, since there is possibility of getting hacked through internet. The IoT devices are

*Is it possible that the prototype implementation can be implied on the current system without much change or no changes in its way of working?*

It is possible and implemented by using a SOC device raspberry pi which will communicate with the PLC and change the tag values. But still there were some additions added to the PLC program to make it work.

The other findings, comparison with the similar works, practical usage, and limitations are discussed below.

Some of the findings which are crucial to the development of a safety system are. The safety system should trigger only when needed. For example when using a laser based safety system in an environment where compressed steam is used, the laser light could be disturbed by the steam. But in the communication based safety there is no issue with the interference of steam. The communication is a short range but it is effective enough to communicate with the neighbor since the clearance distance is 6 meters which is lesser than the maximum range of the ZigBee.

Comparing with the similar works the prototype used here use communication to change tag values instead of connecting with the IO port. Connecting to the IO ports has the limitation of adding extra IO modules and the analog input output could be questionable due to the converter used to step up the voltage to 24 v. By communication based control a specific value can be written in the tags. Using a raspberry pi also gives an opportunity of having a micro server which could be accessed through network and can be used to log, study and control the monorail carrier for diagnostic purpose. Even some ZigBee based wireless sensors can also be attached and the logic can be set in the raspberry pi instead of changing it in the PLC.

Some of the clinical relevance observed in this project which can be practically executable. During a communication failure, it is possible to move the carrier to a specific point by communicating using the secondary communication. Using the IR remote control to move the carrier to the maintenance position for a few tens of meters at a slow speed consumes much time. And also if the IR line of sight is disrupted by any structures the carrier will stop, causing delay in the moving of carrier to the maintenance position and lead to production loss. Human error also occurs like forgetting to replace worn out batteries in the IR remote causing such delays. In the communication model the neighboring carrier will do the task avoiding any human interference.

Some of the limitations are the PLC program which cannot be modified much, in the PLC program the tag values are update during every cycle, the messages sent by our system will change the value only once but in the next cycle its values are overwritten by the PLC. The possible solution for this is to have a separate function block which could be executed at the end of the PLC program cycle. Also one of the limitation is that accident which stopped the further testing different methods to explore the possibilities of controlling the PLC. And also the unavailability of readily available products which can be used with the PLC. For example the ZigBee module used is a developer module which is not a complete commercial product to be used with a PLC.

## 6. CONCLUSION AND FUTURE WORK

This thesis concentrates on the solution which prevents the collision between monorail carriers using a secondary communication as the prevention mechanism. This chapter concludes the project by providing the overall vision about the project. And the possible future works which could be implemented are discussed.

### 6.1 CONCLUSION

The purpose of this thesis is to use IoT based collision avoidance mechanism in a closed loop monorail system. That IoT based collision avoidance system should be a communication based system. During a communication failure in any of the carriers, the neighboring carrier should know the position of the carrier which lost its communication with the floor control. This communication should not interfere with the primary wireless communication which used 5 GHz, so the secondary wireless communication should not be a 5 GHz operating one. The requirements of this implementation is listed out in the introduction of chapter 3.

In this implementation an IoT device raspberry pi is used to acquire data and control the PLC which operates the monorail. The communication between each raspberry pi is done using ZigBee wireless technology which is explained in chapter 2.5.2. The ZigBee uses 2.4 GHz thus will not be an interference. The building is done in 4 stage for achieve the final result which is shown in Figure 24. The secondary wireless communication communicates using message passing sharing its position with all the other carriers. The message structure is explained in chapter 4.3.1. For the prototype the programming is done in the raspberry pi using python and it communicates using ZigBee with other carriers and Ethernet IP to communicate with the PLC. During communication failure the carrier IP, status message, and position of the carrier is broadcasted in the network. If the neighboring carrier is closer to the affected carrier, then it will stop and asks the affected carrier to move to the homing position. By this way the unnecessary delay in the manual handling the carrier to the maintenance position is reduced. The logic is written to imitate the procedures which will be used by the maintenance engineer during troubleshooting.

The controlling of the PLC using the raspberry pi is tested by measuring the time difference before and after reading the value from the PLC. The time taken by the raspberry pi is less than 100 ms making it a real-time communication. The readings taken using the test program is shown in Figure 44. And the ZigBee wireless throughput test and range tests are taken for 1 meter and 6 meters as shown in Figure 40. The coverage was good and the test is discussed in 5.1. The prototype which is developed here can communicate and control the PLC and share the information in the ZigBee network. In order to get the system work perfectly in full speed the stage 4 which is stated in the Figure 24 should be implemented. But that part is taken as the future work. Due to the

accident which happened while testing the program, a simulator which shows the graphical working of the system is built which is discussed in the chapter 4.5.

## **6.2 FUTURE WORK**

The future work is to connect a ZigBee module straight to the PLC using serial to Ethernet converter. MOXA N5000 series can convert the serial data to Ethernet packets and transmit over the network to the PLC. In addition, creating a separate function block in the PLC to handle the logic and communication. Also in the future the ZigBee network could be used to communicate with mini servers which could be used for collecting data and the information about the monorail carriers. It is also possible to have a mobile phone application which can be used to change the tag values in the PLC. Also adding a raspberry pi along with the PLC could open a path by having a mini server which could run services like web browser based user interface. By making a browser based user interface it is possible to have raspberry pi based kiosk type user interfaces which use less resource.

## 7. REFERENCES

- [1] Product Handling Concepts. (2014). The History of Conveyors. [online] Available at: <https://www.phcfirst.com/words-in-motion/2014/6/30/the-history-of-conveyors> [Accessed 19 May 2018].
- [2] Nancy G. Leveson. 1986. Software safety: why, what, and how. *ACM Comput. Surv.* 18, 2 (June 1986), 125-163. DOI=<http://dx.doi.org/10.1145/7474.7528>
- [3] N. G. Leveson and P. R. Harvey, 1983. Analyzing Software Safety. *IEEE Transactions on Software Engineering*, vol. SE-9, no. 5, pp. 569-579, doi: 10.1109/TSE.1983.235116.
- [4] Python EtherNet/IP CIP API (2016, September 20). Retrived from <https://hardconsulting.com/products/6-cpppo-enip-api>
- [5] M. Waidner and M. Kasper, "Security in industrie 4.0 - challenges and solutions for the fourth industrial revolution," 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, 2016, pp. 1303-1308.
- [6] A. Moallim, J. M. Lee and D. S. Kim, "Wireless control and monitoring using Programmable Logic Controller (PLC)," 2017 17th International Conference on Control, Automation and Systems (ICCAS), Jeju, 2017, pp. 1763-1767. doi: 10.23919/ICCAS.2017.8204259
- [7] Ron Schneiderman, "Internet of Things/M2M??-??A (Standards) Work in Progress," in *Modern Standardization:Case Studies at the Crossroads of Technology, Economics, and Politics* , 1, Wiley-IEEE Standards Association, 2015, pp.288 doi: 10.1002/9781119043492.ch9
- [8] M. Mashaei and B. Lennartson, "Concurrent design and control of automated material handling systems," 2013 IEEE International Conference on Automation Science and Engineering (CASE), Madison, WI, 2013, pp. 292-299. doi: 10.1109/CoASE.2013.6654041
- [9] I. Paprotny, Juin-Yan Shiau, Yo Huh and G. T. Mackulak, "Simulation based comparison of semiconductor alternatives: continuous flow vs. overhead monorail," 2000 Winter Simulation Conference Proceedings (Cat. No.00CH37165), Orlando, FL, 2000, pp. 1333-1338 vol.2. doi: 10.1109/WSC.2000.899104
- [10] Xinsheng Che, Yun Sun, Bo Li and Hui Xu, "Design and system performance test of a kind of PLC with wireless communication function," 2017 3rd International

- Conference on Control, Automation and Robotics (ICCAR), Nagoya, 2017, pp. 762-765. doi: 10.1109/ICCAR.2017.7942800.
- [11] M. Hermann, T. Pentek and B. Otto, "Design Principles for Industrie 4.0 Scenarios," 2016 49th Hawaii International Conference on System Sciences (HICSS), Koloa, HI, 2016, pp. 3928-3937. doi: 10.1109/HICSS.2016.488
  - [12] Product Handling Concepts. (2014). The History of Conveyors. [online] Available at: <https://www.phcfirst.com/words-in-motion/2014/6/30/the-history-of-conveyors> [Accessed 19 May 2018].
  - [13] Library.Automationdirect.com. (2018). History of the PLC. [online] Available at: <https://library.automationdirect.com/history-of-the-plc/> [Accessed 19 May 2018].
  - [14] Service, E. and Service, E. (2015). PLC Engineering, Design, Manufacturing, Product Development. [online] Electronic Product Development, Design, Prototype, Manufacturing Service. Available at: <http://electronicdesignmanufacturing.com/blog/programmable-logic-controller-engineering-design-development-manufacture/> [Accessed 19 May 2018].
  - [15] Sick AG. S3000 Safety Laser Scanner: Operating Instructions. Retrieved from <http://sensorsincorporated.com/uploaded/Doc/Sick%20S3000%20Manual.pdf>. 2010.
  - [16] Nordum, Amy. (2018). Popular Internet of Things Forecast of 50 Billion Devices by 2020 Is Outdated. [online] IEEE Spectrum: Technology, Engineering, and Science News. Available at: <https://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated> [Accessed 21 May 2018]
  - [17] Deoras, S. (2018). First ever IoT device- "The Internet Toaster". [online] IoT India Magazine. Available at: <https://www.iotindiamag.com/2016/08/first-ever-iot-device-the-internet-toaster/> [Accessed 21 May 2018].
  - [18] Teicher, J. (2018). The little-known story of the first IoT device - Industries Blog. [online] Industries Blog. Available at: <https://www.ibm.com/blogs/industries/little-known-story-first-iot-device/> [Accessed 21 May 2018].
  - [19] Thehenryford.org. (2018). Monorail Conveyor in Ford Motor Company Highland Park Plant Machine Shop, 1914 - The Henry Ford. [online] Available at: <https://www.thehenryford.org/collections-and-research/digital-collections/artifact/249917/#slide=gs-274819> [Accessed 3 Jun. 2018].
  - [20] Conveyor Image [online] Available at: <https://pxhere.com/en/photo/1292548> [Accessed 3 Jun. 2018].



- [21] Statista. (2018). IoT: number of connected devices worldwide 2012-2025 | Statista. [online] Available at: <https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/> [Accessed 3 Jun. 2018].
- [22] K. H. Kamaludin and W. Ismail, "Water quality monitoring with internet of things (IoT)," 2017 IEEE Conference on Systems, Process and Control (ICSPC), Malacca, 2017, pp. 18-23. doi: 10.1109/SPC.2017.8313015
- [23] Nexcom.com. (2018). IoT Gateway Solution. [online] Available at: <http://www.nexcom.com/applications/DetailByDivision/iot-gateway-solution> [Accessed 3 Jun. 2018].
- [24] P. Li and J. Li, "Application of Communication and Remote Control in PLC Based on ZigBee," 2009 International Conference on Computational Intelligence and Security, Beijing, 2009, pp. 533-536. doi: 10.1109/CIS.2009.32
- [25] Raspberry Pi Model B+. [online] Uk.rs-online.com. Available at: <https://uk.rs-online.com/web/p/processor-microcontroller-development-kits/8111284/> [Accessed 22 Jun. 2018].
- [26] Digi.com. (2018). XB24CZ7WITB003 - Programmable XBee Zigbee Through-Hole (Wire Antenna) - Digi International. [online] Available at: <https://www.digi.com/products/models/xb24cz7witb003> [Accessed 22 Jun. 2018].
- [27] Pepperl+Fuchs. (2018). Position Encoding System (WCS). [online] Available at: [https://www.pepperl-fuchs.com/global/en/classid\\_822.htm#](https://www.pepperl-fuchs.com/global/en/classid_822.htm#) [Accessed 29 Jun. 2018].
- [28] Murakami, K., Ishida, Y. and Okubo, (1976). A.c. relay system. Patentee K. Sinfonia Technology Co Ltd (US patent US4074333A).
- [29] Kauhanen, Eetu. (2017). An approach to autonomous collision avoidance in a Monorail transfer system. M.Sc. dissertation, Tampere, TUT.
- [30] The Finnish Communications Regulatory Authority (2018), Frequency allocation table (annex to regulation M4X), Retrived from: <https://www.viestintavirasto.fi/en/steeringandsupervision/actsregulationsdecisions/regulations/radiofrequencyregulation4.html>

## **APPENDIX A: DIN MOUNT FOR THE RASPBERRY PI AND ZIGBEE**



**Zigbee DIN mount  
3D printed**



**Raspberry pi DIN  
mount 3D printed**

## APPENDIX B: TAG READ TIME CALCULATION PROGRAM

```

from cpppo.server.enip import client
import sys
import time
print(" ")
print("Output:")
timeout = 10
port = 44818
speed = 0
i = 0
f = open('reading.csv', 'w')
# IP = raw_input("Enter the IP address of the PLC: ")      #"192.168.80.121"
IP = "192.168.80.101"

def TagData(mode, tag_name, plc_addr):                      # Building a function
    with client.connector(host=plc_addr,
                          port=44818,
                          timeout=timeout) as conn:         # Creates a UDP
        connection
        operations = client.parse_operations([tag_name])
        if (mode == "wr"):
            failures, transactions = conn.process(operations=operations,
                                                    depth=2,
                                                    multiple=0,
                                                    fragment=False,
                                                    printing=False,
                                                    timeout=timeout) # Write
Tag
        elif (mode == "rd"):
            for index, descr, op, reply, status, value in
                conn.pipeline(operations=operations,
                              depth=2):    # Read tag
                poz = (value[0])
                if value is None:
                    print("None returned while reading %s from PLC %s " %
                          (tag_name, plc_addr))
            return poz

def PushButton(btag, secs): # Function to trigger a value and reset in a
                             # specific interval, like a push button.
    TagData("wr", str(btag)+"=(BOOL)True", IP)
    time.sleep(secs)
    TagData("wr", str(btag)+"=(BOOL)False", IP)

def read(tag):            # Function to read a tag value.
    tagout = TagData("rd", tag, IP)
    return tagout

def wuint(btag, val):     # Function to write a DINT value to a tag.
    TagData("wr", str(btag)+"=(DINT)" + str(val), IP)

def wbool(btag, state):   # Function to write a BOOL value to a tag.

```

```

    TagData("wr", str(btag)+"=(BOOL)" + str(state), IP)
# print (int(round(time.time() * 1000)))
# print (read("SYS_CommOK_CC.DN"))
# print (int(round(time.time() * 1000)))
# print (read("X_ExtEncoderPos"))
# print (int(round(time.time() * 1000)))
f.write("comm;position1;position2;diff of pos;timedif1;timedif2;timedif3 \n")

while True:
    timeon = int(round(time.time() * 1000)) # time in milliseconds
    posit = read("X_ExtEncoderPos")         # read the tag
    timeto = int(round(time.time() * 1000)) # time in milli seconds
    print ((str(posit1) + " ; "+ str(difference) + "\n"))
    difference = (timeto - timeon)
    data = (str(posit1) + " ; "+ str(difference) + "\n")
    f.write(data)

```